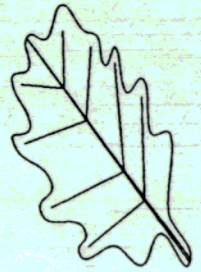


Egebladet

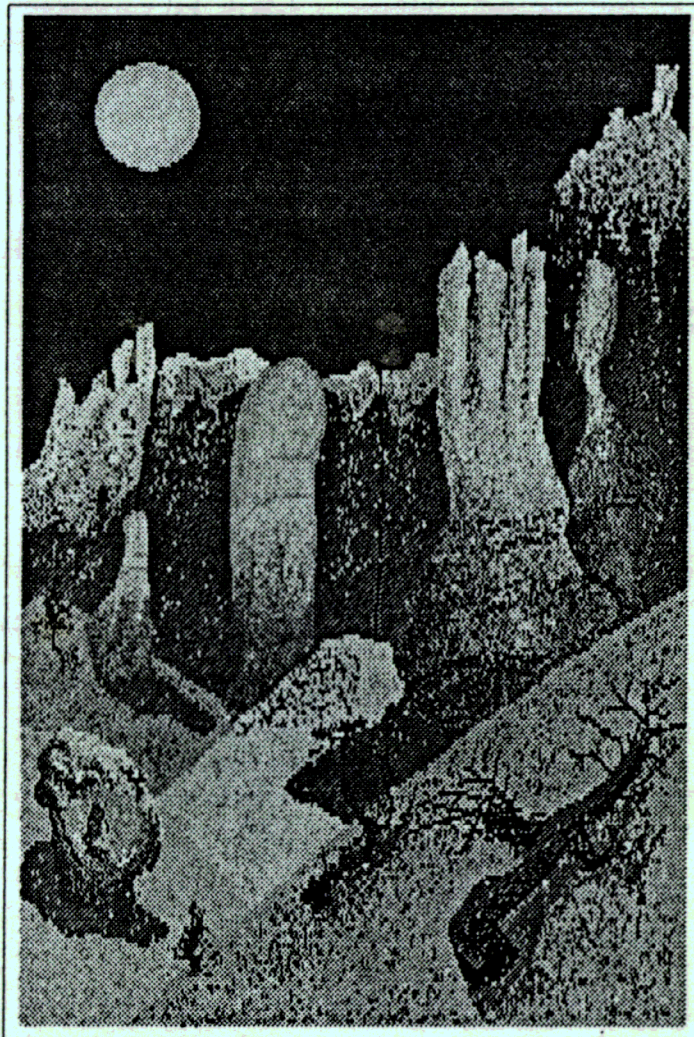


Dansk blad for BBC/Acorn computere

Nr.16

JUNI-JULI

1991



Egebladet

Egebladet er foreningen "QUERCUS" klubblad.

Bladet udsendes til foreningens medlemmer ca. 6 gange årligt.

Redaktion:

Chef-redaktør/

Layout : Olav M.J. Christiansen

Tryk : Stationen, Nørresundby

I redaktionen

i øvrigt : Martin Søndergaard
Karsten Jensen
Brian Madsen
Hans Otto Lunde

Adresser:

Indlæg til *Egebladet* sendes til nedenstående adresser.....

Adventuresiden/Klubside/BBC-spil

Martin J Søndergaard
Årestrupvej 76
7470 Karup J

Programmering etc. BBC

Thomas Olsson
Kagså Kollegiet 155/5
2730 Herlev

Programmering etc. Archimedes

Jens H Ovesen
Tolderlundsvej 27, 4tv
5000 Odense C

Tips&Tricks/Anmeldelser/Annoncer/

Læserbreve og alle andre bidrag til bladet

Olav M J Christiansen
Sundvej 30, Hvalpsund
9640 Farsø

Husk at mærke kuverten "Egebladet".

Artikler og længere læserbreve må meget gerne afleveres som ASCII fil på 3.5"/5.25" disc. Disketter returneres, selvfølgelig.

quercus

Dansk Acorn og BBC Bruger Gruppe

Formand Olav M J Christiansen
Sundvej 30
Hvalpsund
9640 Farsø
☎ 98 63 81 50

Næstformand Thomas Christiansen
Kildebakkegårds Allé 42
2860 Søborg
☎ 31 67 89 00

Kasserer Martin J Søndergaard
Årestrupvej 76
7470 Karup J
☎ 97 10 21 95

Quercus BBS ☎ 31 67 97 70

SysOP Næstformanden

Henvendelser vedrørende klubben/
adresseændringer rettes til kassereren.

Indbetalinger til klubben:

Giro 4 36 71 54
QUERCUS
Dansk Acorn og BBC Brugergruppe
Årestrupvej 76
7470 Karup J

Egebladet nr 16 juni/juli 1991

Redaktionen sluttet 23 juli 1991
Deadline for nr 17 31 august 1991

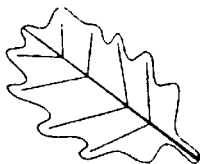
Forside:

"sakset" fra Enthar 7
(igen)

Boni Balani Credo

Indhold

Nyheder	4
Udklip fra de seneste nyheder angående Acorn/BBC-computere	
Højt at flyve	5
Nick Etlar Eriksen tager os op at flyve.	
BBS PD-software	8
Thomas SysOp fortæller om Public Domain software på Quercus BBS.	
Emula2eren	9
Seneste nyt om den nye PC-emulator til Archimedes/A3000.	
Generalforsamling 91	11
Så er det tid for den årlige komsammen igen.	
DisARMing	12
Dette gang fortæller Hans Otto blandt andet om neurale netværk	
Programmering	26
Ny artikelseerie om programmering.	
EDB på arbejdspladsen	28
Kay sammenligner Mac med Arc.	
Sideways RAM til BBC	33
Lars Østerballe fortæller hvordan.	
Cambridge-Pascal	35
Anmeldelse.	
Adventuresiden	37
Martin Søndergaard guider os det stidste stykke igennem Enthar 7.	



Redaktionelt

Kære læser.

Velkommen til endnu et fantastisk nummer af Egebladet(!). Jeg håber ikke, folk har opgivet Egebladet, selv om det engang imellem udkommer lidt uregelmæssigt.

Atter engang har vi en blandet landhandel; der er artikler om sideways RAM, om PD-software, spilmeldelser og meget mere. Der er dog for en gangs skyld ingen læserbreve denne gang. Skal man tolke det sådan, at læsere er tilfredse? Eller var der bare for kort deadline ("snarest mulig")? Det ved kun læsere.

Der er jo desværre ikke udkommet så mange Egeblade her på det sidste, som vi havde håbet på, men til gengæld er dette nummer så blevet ekstra stort. Hele 40 sider er det blevet til, men bladrer man det igennem, vil man konstatere at det meste af det stadig er produceret af de samme mennesker. Hvis I derude i det danske land er interesseret i at bevare Egebladet og prøve at holde en rimelig jævn udgivelsesfrekvens, må I altså til at indsende mere stof.

Det kan selvfølgelig også bare være, at det er agurketid, hvad indlæg angår, men jeg vil dog lige gøre opmærksom på, at der ikke er ret lang tid til næste blad udkommer, idet vi regner med at have det ude lige efter generalforsamlingen.

OBS: Læg venligst mærke til at GF91 er flyttet! Vi måtte desværre i sidste øjeblik pga. tekniske vanskeligheder (den undskyldning plejer TV at bruge, så det kan vi vel også) udskyde generalforsamlingen 14 dage - til afholdelse søndag den 18. august. Dette har også betydet, at det er blevet muligt at afholde GF91 i København, som mange har ønsket. Se nærmere inde i bladet.

Er der ivoirigt nogen af jer derude, der kunne tænke jer at være med i redaktør-juntaen? Vi kunne godt bruge et hoved' eller to mere, sådan at vi kan arbejde lidt på skft.

Mvh. Redaktøren (Olav)



Nyheder

Stort og småt fra Acorn- og BBC-verdenen

Atter engang bærer nyhedsspalten meget præg af Archimedes/A3000. Det må desværre indrømmes, at der ikke sker meget nyt mht. BBC/Master m.v. Hvis der er nogen ude i det danske land, der hører noget interessant, der angår BBC-computere (også selv om det kun er rygter) skal de være velkommen til at skrive herind (se adressen foran i bladet). Nyheder, der angår alle andre Acorn computere, har selvfølgelig også interesse. Måske er der nogen, der har lyst til at være 'informere' for bladet?

BBC Acorn User Show

Nu er det snart tid at sætte kryds i kalenderen, hvis man skal med til det store årlige show for BBC/Acorn-computere. Årets show finder sted fra den 11. til den 13. oktober og afholdes i Londons Wembley Conference Centre. Af ting, som Acorn muligvis har tænkt sig at vise frem på dette show, kan nævnes en bærbar Archimedes. Denne har været på tegnebrættet længe, og det er meget sandsynligt, at den bliver udsendt i år. En anden ting, som mange brugere har ventet længe på, er det nye RISC OS 3. Det har allerede verseret længe i form af forskellige moduler, men den færdige version forventes at indeholde meget mere end de beta-versioner, der har været i omløb.

Ny PC-emulator

Acorn har nu endelig udsendt den officielle version af den nye PC-emulator (version 1.60). Prisen skulle være ca. £99 og eksisterende brugere kan opgradere for £29. Se også Emula2eren i dette nummer.

RiscIx frigivet

Acorns version af UNIX, RiscIx, tidligere kun tilgængelig på R160 og R500-serien, kan nu købes i løs vægt. Prisen vil dog nok afholde de

fleste privatpersoner fra at anskaffe sig den. For Archimedes brugere er prisen fra £999 (eller omregnet til danske kroner ca. 14.000 inkl. moms) for at få den seneste version af systemet (inkl. X Windows System version 11 release 4). Eksisterende brugere af RiscIx på R140 kan upgradere til det seneste system for £349.

Nyt SCSI-kort fra Acorn

Det nye SCSI-interface fra Acorn er nu endelig blevet udsendt, efter flere måneders ventetid. Den medfølgende software virker dog ikke helt færdig. F.eks. overfører fileren kun i blokke af 1K, og dette gør kopiering i desktop'en ret langsom. Overførselshastigheden er iøvrigt omkring 1.25 MB/s.

Nyt terminal program

ECD Computers Delft, i Holland, har netop udsendt et nyt terminal-program, specielt beregnet for Viewdata (MODE7). Programmet er multitasking og indeholder bl.a. en Teletekst-editor, faktisk en slags tegneprogram. Endvidere supports både ANSI, VT og TTY.

Nye spil

Sammen med et racerbil spil og bil-simulator på én gang fra Impact Software.

Fra samme firma kommer Boogie Buggie (nej, spørg mig ikke - men det skulle være noget med at styre en 'flying hero').

Nyt video-kort

PCATS 24 bit video enhancer board er et nyt kort, der giver ægte 24 bits palette. ROM'en indeholder de mest populære skærmmoder, inklusive Computer Concepts'.

Højt at flyve...

Nick tager os med op at flyve i henholdsvis Interdictor II og Chocks Away v.2

INTERDICTOR II af Clares

Simulationsspil er meget populære hos masserne, og med god grund fordi de er virkelig noget særligt. Des hurtigere computer, des bedre simulationsspil og det er lige netop af den grund at det er oplagt at lave sådanne programmer på en Archimedes. Nogle af de allerbedste spil til Archimedes er da også simulationsprogrammer. Nu sker der omsider også noget på vores front, Interdictor II, Chocks Away (v.2) og Powerband er de seneste.

Interdictor II er efterfølgeren til Interdictor I. Om der kommer en Interdictor III, som så vil være efterfølgeren til Interdictor I's efterfølger; Interdictor II, vil tiden vise, men vi kan da håbe. Interdictor-serien er alle fly simulatorer - altså noget der skulle efterligne effekten af selv at flyve et fly.

I Interdictor II har man kontrol over et moderne jettfly med varmestyrede missiler, gun og raketter - altså som i Interdictor I. Der er dog sket ændringer; De kedelige prikker som repræsenterede raketter i Ieren er blevet erstattet af en mere attraktiv ild-udflåd bestående af splinter og streger der giver en flot perspektivvirkning der får dem til at se lidt farligere ud end den lille håndpistols spredte prutskud.

Udover våbensystemet og gameplayet er næsten alt andet ændret. Det tynde, spæde landskab fra Ieren er blevet erstattet af et stort kvadrat fyldt med godter som jernbane og tilhørende tog med vogne, radarstationer, bittesmå byer med kirke, motorveje med lastbiler, åbne biler og andre biler samt et kæmpestort krigsskib med tilhørende sø og et fly der smider kasser ned med faldskærme, der når de lander, forvandles til...(last)biler. Stadigvæk er der ultrarealistisk simulering af andre jagerfly der skal lande og tanke op, skud der eksploderer når de rammer, men som en

tilsætning, at hvis man rammer nogen jagere med missiler, eksploderer de og får en røghale medens de går i spind og til sidst braser ned for at baldre forhjulene mod jorden. Simuleringen af eget og andre grafiske objekter er overdådig. Flyene har fået lidt mere farve, ser uhyggeligt godt ud og er mange flere end i Ieren. Personligt har jeg selv været i nærkamp med 8 på een gang! Man kan nu kigge ud af ikke kun forrude, bagrude(?) og siderne, men også de fire mellemstillinger -altså i alt 8 retninger.

Men for nu at se på de faktiske forbedringer fra første version, kan følgende fremhæves: Jeg savnede noget grafisk orientering om hvor langt fra jorden man var. Dette er blevet rettet i Interdictor II, der er altid streger eller marker som sørger for elimineringen af unødvendige og irriterende nedstyrt. Et stort hurra. Styrerutinen er stadig med mus, men er efter min mening blevet en anelse bedre. Jeg har intet mod at styre rundt med en mus, men muligheden burde være der for evt. styring ved hjælp af tastaturet. Man letter stadig for hurtigt, den lækre følelse som det er at lette, er endnu ikke helt i orden. Hastigheden er ikke faldet, de har tilsyneladende formået at udnytte ARMens specielle features. Der er stadig mulighed for umiddelbar installering på Harddisk. Lyden er ikke ændret, og det samme gælder selve gameplayet - nok er landskabet blevet anderledes (til det bedre), men det drejer sig stadig om at erobre landingsbaner. Beliggenheden af disse kan man se på et stort kort inden man starter spillet.

Inden man starter spillet kan man også vælge at spille udødelig, hvilket simpelthen betyder at man kan øve sig lidt uden at være bange for at styrte ned og skulle begynde forfra. En god ting som mange flysimulatorer tilbyder. Et minus er stadig at det ikke er muligt at lave "spot" på flyet, men vi håber at det er med hvis der kommer en IIIer.

Kan man lide simulationsspil er dette en klar 

køber, og kan man ikke, så køb det alligevel da det er ét af de allerbedste spil til Archimedesen på markedet på nuværende tidspunkt. For fuldstændighedens skyld, så bedes du (De hvis De har et identitetsproblem) læse min anmeldelse af Interdictor I i sidste nummer af Egebladet. Karakterer og vurdering følger efter sammenligningen med Chocks Away.

Chocks Away (v.2) af Fourth Dimension

Clares er imidlertid ikke den eneste udbyder af flysimulatorer. Fourth Dimension er heldigvis også hoppet med på vognen. Lidt konkurrence tvinger kvaliteten op og prisen ned, så lad os alle feste og klappe i vores små fedtede hænder.

Som et modstykke til Clares top-moderne jettfly, tager Fourth Dimension os tilbage til tiden hvor en lille lækker rød dobbeltdækker med maskingøb var vindens ubestridte hersker. Desværre er der ikke mulighed for at skyde prøjserne til blutwurst (som vi har gjort det før et par gange), så den fornøjelse må vi gemme til en anden god gang. Men Chocks Away har mange andre features. I main-directoriet er der mulighed for at vælge enten at spille selve spillet, noget som hedder EXTRAS eller se en demo. Derudover at formatere en BLACK-BOX disk eller, hvis man har en A3000, nogle andre taster som skulle sikre at spillet ikke crasher undervejs.

Hvis man vil spille, loader man det ind, og får en menu der tilbyder at øve sig lidt på op til 3 forskellige (store firkantede) landskaber som hver især har mange spændende ting at udforske. En af dem har sågar et hangarskib som ens egen hjemmebase - den man skal lande på når man har gennemført en mission. Der er ialt 20 missioner som skal udføres for at gennemføre spillet. Missionerne går alle ud på at man, dels på et kort og dels skriftligt, får specificeret et mål - typisk at man skal skyde de og de fly ned, ødelægge et kontroltårn, baser, tanks, bombefly med eskadriller o.s.v. En god ide med disse missioner da det giver en smule variation i gameplayet. EXTRAS er en slags combat-øve-simulator, idet man bliver sat (midt i luften) i en kampsituation, omgivet af fjender. Om du så vil snakke eller skyde først, er op til dig. Man har både i selve spillet og i combat-øve-simulatoren mulighed for at spille two-player, og på een gang! Således kan man spille sammen på missioner eller bare gøffe lidt,

eller/og påbegynde en udmattende dog-fight indbyrdes. Endvidere kan man formatere en BLACK-BOX disk, sætte RECORDER ON og derefter ellers bare flyve derudad. På denne måde gemmer man sine flyveture og luftkampe på disk, hvorefter det er muligt at REPLAY - afspille den igen (ialt 382 minuters fuldstændig identisk afspilning!).

Grafisk er der en del forskellige slags fly, tanks og skibe, broer, veje, fodboldbaner, floder, søer o.s.v. På jorden er der prikker som viser hvor langt man er nede, men de er rimeligt store, og man har ikke vanskeligheder med at se hvornår det går galt. Stadigvæk er det lidt nemmere med streger som i Interdictor, og streger ser også lidt bedre ud. Men i Chocks Away kan man vælge at se flyet med spot på, altså se flyet foran sig, fra siden eller forfra. På denne måde kan man se skygge når man nærmer sig jorden. Det ser ufatteligt godt ud, men af og til kan det vise sig ikke at være så smart at benytte sig af dette spot når det bliver hedt omkring sig fordi man ikke kan se sigtekornet, hvilket gør det problematisk at ramme. Grafisk er der også nogle formidable dø-sekvenser, man skifter vinkel, ser sit fly fra siden mens det, med røg i halen, drøner mod jorden for at ende i en flot og farverig (næsten 100%) vektor-eksplosion. Dette gælder for alt man skyder ned. Desværre er denne flysimulator ikke helt spritefri idet røgen og lidt af eksplosionerne også består af en række små sorte sprites. Det ser ganske godt ud, men er ikke helt legitimt for et vektorgrafisk program.

Lyden er en sagte prutten, en lyd flere af os sikkert er bekendt med, men som er ganske god af en flysimulator at være. Lyden er iøvrigt ikke særligt varieret, kun een slags boom når et fly laver nærkontakt med jorden og maskingeværets hidsige krasssen. Der var en Chocks Away v.1 (den første), og den var "uudholdeligt" langsom. Denne version som er anmeldt (v.2) har fået lidt mere grafik og en kraftig doping, men stadigvæk er den ikke så flydende som Interdictor II. Dette er ikke fordi der er mere grafik på skærmen, men har en anden årsag - måske at Interdictor først får de allersidste detaljer med (som for eksempel bjælderne over skinneme) lidt senere end den slags kommer på Chocks Away. Dog skal det lige nævnes at Chocks Away har en helt og aldeles acceptabel hastighed, det skal man ikke lade sig skræmme af.

Med hensyn til simulering er den ikke så omfattende sammenlignet med Interdictor II, kun simulering af fly (dog en del forskellige), og tanks. Simuleringen er ganske god, men til tider føler jeg at den ikke er helt perfekt. Der er ikke nok inertie selv om, selvfølgelig, at de små to-dækkere kunne lave nogle mærkværdige drej og vendinger som en nutidig jager simpelthen flyver for stærkt til at kunne gøre efter. Muligvis er det lige netop det som gør den lille røde sjov at flyve.

Med hensyn til specielle features er Chocks Away, Interdictor II overlegen, two-player option, recorder, flere landskaber, spot og mere. Det skal programmørerne af Chocks Away roses for, og det gør nok i sidste ende lidt til at interessen bliver længere hos Chocks Away og kortere hos Interdictor II. Ros: det var vel nok rigtig flot og fint. Men for begge gælder det, at det er den slags spil man altid vil tage frem igen engang. Selv om man gennemfører spillet ser det jo stadig uendeligt godt ud at fare rundt over landskabet i 3D.

Et sidste afsluttende plus må være muligheden for at styre flyet med taster, joystick og mus -altså helt op til hvad man selv foretrækker. Mange tak for det! En meget afsluttende kommentar skulle da lige være at Chocks Away muligvis er en anelse for let, men Fourth Dimension har lovet en extra disk med nye godter, så måske er det ikke så dårligt endda, for på den anden side skal et spil jo heller ikke være for svært!.

Interdictor II		Chocks Away (v.2)	
Grafik	11	Grafik	11
Hastighed	11	Hastighed	9
Lyd	7-8	Lyd	7
Simulationsværdi	11	Simulationsværdi	10
IQværdi	8	IQværdi	10
Interesseværdi	9-10	Interesseværdi	10-11

*IQværdi hentyder til de muligheder et spil tilbyder, hvor forudseende programmører har været, og i hvilket omfang de har taget hensyn til spilleren.

Små nyheder

(news in brief)

Scanlight Junior256.

Videreudvikling af CC's Scanlight Junior. Den kan klare 256 eller 16 gråtoner. Opløsningen er 400 DPI.

Nye fonts.

Der bliver ved med at komme nye outlined fonts til Archimedes/A3000. Longman Logotron har netop udsendt Morse, Braille og Semaphore!

Nyt scanner-interface.

Clares Scanpack er et podule, der kan anvendes sammen med en Epson GT400 farve scanner. Denne scanner i 24 bit farver eller 256 gråtoner og en opløsning på 400 DPI.

Ny program-pakke fra Acorn

En ny udviklingspakke med 19 c-programmer og 17 assembler do. skulle gøre det nemmere at udvikle software til RISC OS. Inkluderer forskellige værktøjer, bl.a. til debugging.

ANNONCE

Sælges:

32K shadow RAM

(fra Watford)

Pris: kr. 200,-

Henv. : Martin 97 10 21 95

BBS PD-software

Thomas SysOp fortæller om nyt software fra vores BBS

Noget af bestyrelsen (dvs. Olav MJC og undertegnede) er blevet enige om at vi i fremtidige numre af Egebladet vil bringe en kort beskrivelse af de nyeste programmer på BBS'et. Dels for at vise hvad folket går glip af ved ikke at have et modem (!), og dels for at fortælle lidt om hvad der sker i Archimedes-verdenen udenfor lille Danmark.

Menon, v.1.86. En 'alternativ' filer hvor man selv kan opbygge sine menuer som man vil have dem. En meget overskuelig måde at arrangere sine mest brugte filer/programmer på. Et must for folk med harddisk! Applikationen er skrevet af hollænderen Joris Röling, der allerede nu er et 'etableret' navn indenfor PD-software til Archimedes.

IDEtest. En ganske ubrugelig, men meget morsom applikation der kan teste ens harddisk i alle leder og kanter. Man får desuden mulighed for at save resultaterne så man kan sammenligne med andre diske.

VKiller230. Viruskiller v.2.30, skrevet af Richard K. Lloyd - også en 'etableret' gut indenfor PD. Virus, hvad er nu det for noget? spørger folk.... Jooh, der er i den senere tid desværre nogle vittige hoveder der er begyndt at lave virusser til Archimedes - endnu har de ikke været 'fatale', dvs. slettet programmer og lignende, men det er nok kun et spørgsmål om tid før vi også ser sådan een! VKiller2.30 scanner efter 7 forskellige vira. God at have hvis man har meget PD-software, da det som oftest er den måde de spredes på!

PTC, Pascal to C-'pakke', forfinet af Tor Lillqvist og Graham Toal. Sikkert uundværlig hvis man skal konvertere programmer fra Pascal til C (hvilket man jo ikke ligefrem kan sige at jeg gør dagligt !!). Applikationen er 335 Kb stor, så _noget_ må den jo kunne :-)

GrapeVine v.1.32. Shareware kommunikationsprogram til Archimedes. Programmet supporterer 4

terminaltyper: Ascii (TTY), PetAscii, AtAscii (Atari ??) og ANSI. Filprotokoller er X- og Y-modem. Programmet kan også håndtere dial-directories og desuden medfølger en uddybende manual. Programmet kører dog ikke multitasking, hvilket begrænser dets brug i desktoppen en del! Mig bekendt findes der dog endnu ingen PD commsprogrammer der kan køre multitasking.

FarBeyond. Et ganske morsomt lille spil á la tetris. Spillet er virkelig fængende og kører (naturligvis ?!) multitasking.

Fastbrot3. MODE21 mandelbrots. Programmet ligner til forveksling alle andre mandelbrot programmer for folk uden ARM3, men med denne går det virkelig stærkt. Startbilledet tegnes på ca. 4 sekunder... i MODE21 !! Grunden til denne hastighed skyldes at selve programmet ikke fylder mere end ca 6 Kb, hvilket bevirker at det meste kan ligge i cachen. PC-ejere og Amiga-folk går i coma når de ser det, og så er det ikke en gang løgn !

RiscDream. Armaxess demo nummer 2. Nok en af de bedste demoer til Archimedes overhovedet. Den består faktisk af 3 forskellige demoer der køres sekventielt efter hinanden. Mens demoerne loades ind og udpakkes vises en lille stump grafik der for bare et år siden ville have været en demo i sig selv ! Specielt demo3 er imponerende, her vises en 'landscape flyover' hvor man siddende i et cockpit flyver rundt i et landskab spækket med vektorbaserede objekter, bla. en A3000! Det morsomme er, at de fleste objekter bevæger sig - realtime!

MTV. Mark Terrence VandeWetters raytrace-program. Også en heftig sag på over 200 Kb. Programmet er portet fra Unix, og kører derfor ikke multitasking. Et andet der gør dette er QRT (Quick RayTracer). Hvor quick den er kan diskuteres. Et MODE21 billede tager ca. 8

timer at trace, men så bliver det til gengæld også flot!

DrawPlus. Dette er Draw version 2.00 - en absolut forbedring i forhold til Acorns 'originale' Draw. Det morsomme er at programmet ikke kommer fra Acorn, men fra Jonathan Marten -en engelsk PD-programmør. Af forskelle fra Draw kan nævnes: Op til 32 lag i hver tegning; objekter kan låses; symboler eller clip-art kan gemmes i libraries samt 10 forskellige linie-mønstre. Dette er kun et lille udpluk - der er mange andre forbedringer. For folk der kender og bruger Draw er DrawPlus et must!

SCSI_187. Version 1.87 af Lingenuity's SCSI-software til deres SCSI-podul. Denne version indeholder flere forbedringer og går ikke ned så tit som de gamle versioner gjorde (DET er noget jeg kan tale med om !!). Desuden supporter den flere partitioner på samme harddisk.

new_PrinterPS/LJ/Lx/DM. Officielle nye printerdrivere fra Acorn. Væsentlig bedre end de gamle; man kan blandt andet få sin printer til at skrive ud i NLQ eller draft-mode meget nemmere end med de gamle drivere.

Creator113. John Kortink's nye ShareWare version af MakeGIF og MakeTIFF. Denne applik-

ation kan lave sprites om til GIF, TIFF og AIM-format.

Transla645. Seneste version af John Kortink's Translator; denne er nu også ShareWare! Translator er en pendant til Roger Wilson's ChangeFSI - men meget bedre efter min mening. Translator kan læse praktisk taget alle billedformater og konvertere dem til sprites. Uundværlig!

Sidst men ikke mindst kan jeg oplyse at der på BBS'et nu ligger ca. 4,5 Mb outlined fonte, fordelt på næsten 100 forskellige typer, samt ca. 7 Mb SoundTracker tunes; de resterende ca. 25 Mb ligger off-line på floppy'er. Jeg har en komplet liste over alt hvad vi har af SoundTracker tunes liggende til download for de interesserede. Pr. 30/06/91 fylder download-sektionen ca. 69 Mb.

Med venlig hilsen *ThCh - SysOp*.

P.S. Jeg skriver dette mens jeg lige er hjemme for en kort bemærkning - på tirsdag (d. 02/07) drager jeg af mod Korfu. Men i de forløbne 9 dage hvor jeg ikke har været hjemme, har BBS'et haft 260 opkald - og det gik slet ikke ned !!!!! Samtidig i mit fravær, gik Ib Michael Martinsen hen og blev den heldige (!) der ringede til BBS'et som nummer 10.000 (ti-tusind !) - så skal vi bare finde en præmie til ham ;-)

Emula2eren

Den nye PC-emulator er på vej...

Så er emula2eren tilbage igen for en kort bemærkning. Anledningen er den længe ventede udsendelse af Acorns nye PC-emulator til Archimedes/A3000. Jeg har desværre endnu ikke set den færdige version. Den, jeg har afprøvet, er en Beta-version (2.58). Denne er dog rimelig bugfri, så jeg vil formode, at den endelige version kommer til at ligne den.

Det nye er, at man nu kan køre PC-emulatoren i baggrunden - altså under RISC OS. Når man klikker på !PCem icon'en installerer den sig på icon-bar'en, og man kan nu vælge mellem

følgende menu-punkter:


Info

Single task (gråt)

Freeze (gråt)

Configuration

Quit

Single task og Freeze kan dog ikke vælges før man har startet selve emulatoren op. Dette 

gøres ved at klikke på icon'en. Før man gør dette, kan man dog vælge at ændre konfiguration. Dette gøres (surprise surprise) ved at klikke på Configuration i menu'en. Configuration har sit eget RISC OS vindue, hvor man kan vælge forskellige ting: RAM størrelse, floppy disk drev, hard diske og display adaptorer. Dette skal gemmes i en konfigurations-fil før det kan anvendes. Af display adaptorer kan man vælge mellem følgende: CGA, EGA, EGA+ OG MDA. For EGA kan man endvidere vælge mellem almindelig farveskærm eller "enhanced" farveskærm (f.eks. multiscan monitor). Man kan endvidere vælge hvor meget RAM EGA-kortet skal bruge, 65K, 128K eller 256K. EGA+ bruger maksimum. For MDA kan man vælge font størrelse. Der er ivoigt også mulighed for at indtaste en start tekst, som PC-emulatoren skal skrive, når den starter op, men hvad man skal bruge det til, aner jeg ikke. Det er ret simpelt at lave dette i AUTOEXEC.BAT-filen!

Den nye emulator er ret sulten, hvad hukommelse angår, idet man skal have mindst 2MB RAM. Og selv med 2MB kan man ikke køre de højeste EGA-modes, men med 4MB går det meget fint. Mens jeg sidder og skriver denne artikel (i !Edit), har jeg lige startet !PCem op i EGA+. Den bruger nu 1824K og da jeg har en RAM-disk og nogle få andre små applikation kørende, har jeg "kun" 192K fri! Bortset fra det, giver det ikke de helt store problemer. Hvis !PCem får problemer pga. mangel på hukommelse, kommer den selv ud og fortæller det ("Reducing display to black and white"). Det gjorde den f.eks. lige nu, da jeg gik ind i mode 21 (256 farver).

Når emulatoren er startet op, kan man vælge at Freeze den, dvs. parkere den midlertidigt, enten ved at klikke på close-icon'en i vinduet eller ved at vælge Freeze i menu'en. Det sidste vil ikke lukke vinduet, men kun forhindre processen i at bruge tid. Man kan dog ikke frigive hukommelse på nogen af måderne, idet man når som helst kan starte emulatoren op, der hvor man slap.

I brug fungerer emulatoren nogenlunde som den gamle - den kører dog noget langsommere, når man kører den i desktop'en (og tager meget proces-tid). Man kan dog vælge at køre den i såkaldt Single task mode ved at vælge dette fra menu'en. !PCem vil nu opføre sig stort set som den gamle emulator, dvs. slukke for desktop'en og

overtage hele skærmen, blot med den forskel at man nu kan vende tilbage til desktop'en (og alt hvad man har forladt der) blot ved at trykke på menu-knappen på musen (den midterste knap). Meget praktisk! Den gamle emulator kunne man nemlig kun forlade ved at resette maskinen. Efter sigende skulle emulatoren i single task mode være lidt hurtigere end den gamle version, men det er nok mest ARM3-brugere, der vil kunne mærke dette.

Hvis man kører i desktop mode, har man ivoigt adgang til en ekstra menu, hvor der er følgende to punkter:

Save screen

Connect mouse

Save screen anvendes til at gemme hele PC'ens skærm som en sprite-fil. Det må absolut være den simpleste måde at oversætte et hvilket som helst PC-billede til Archimedes format. Connect mouse gør at Arc'ens mus bliver forbundet til PC'ens mus, hvis man har en muse-driver. Til KayJay (og selvfølgelig alle andre: Det kan sagtens lade sig gøre at bruge mus under PC-emulatoren (også den gamle version 1.33). Eksempelvis fungerer Microsoft Mouse Driver Version 6.10Z upåklageligt. Med den nye emulator er det altså blevet meget nemmere at "koble" sig på musen og ved et tryk på menu-knappen bliver man hængt af igen og er tilbage på RISC OS's pointer.

Konklusion

Jeg vil ikke prøve at give den nye emulatorer karakterer eller noget i den stil, for dette skal ikke være en anmeldelse; men blot en omtale. Man må dog konkludere, at det er et stort skridt fremad fra den gamle emulator. Nu mangler vi kun ægte VGA, SuperVGA, XGA... og hvad med support for 286, 386, 486? Nå, man må vel stoppe et sted.

Pris: £99, men eksisterende brugere kan opgradere for £29.

Sidste nyt om Karma.

Det længe ventede spil er åbenbart blevet yderligere forsinket. Det forventes nu at udkomme omkring juletid, og da i flere omgange.

Generalforsamling

Meddelelse til hele Egestammen

Hermed bekendgøres det, at tiden er inde for den årlige komsammen, også benævnt: Quercus' ordinære generalforsamling 1991 (herefter benævnt GF91).

Der har været lidt problemer af forskellig art med at få den afholdt i år. Egentlig ville vi have holdt den i foråret, som vi plejer, men der kom desværre en del i vejen. En enig bestyrelse flyttede så tidspunktet til efter sommerferien, men desværre skete der en del misforståelser - tildels fordi flere havde ønsket at afholde GF91 i København, så derfor måtte vi flytte datoen (i sidste nr. stod der den 4. august). Kære venner: I får jeres vilje; vi afholder rent faktisk GF91 i København. Den helt præcise adresse er:

Hans Otto Lunde

(det er den nydelige unge mand, der giver os husly)

Bustrupgade 7, 1.th.

1737 København V

Tlf. 31 21 05 57

og tidspunktet er: søndag den 18. august kl. 1300.

Planen er at vi (som sædvanlig) mødes aftenen før for at afholde den uformelle del af generalforsamlingen (deltagelse ikke obligatorisk - men absolut morsom).

Ring og snak med Hans Otto om evt. overnatning. Jyder og andet godtfolk kan evt. ringe til undertegnede om transport (jeg kører nok derover om lørdagen).

Iøvrigt er planen om søndagen nogenlunde som følger:

kl. 0900 Log on

kl. 0930 Morgenmad (for de vågne)

kl. 1000 Alm. samling (opvask)

kl. 1130 Frokost

kl. 1300 Generalforsamling

kl. 1500 Demo/show-tid

kl. ???? Shutdown

Dagsorden for GF91 er ifølge vedtægterne, men da der ikke står noget om denne i vedtægterne, har jeg lavet følgende dagsorden (der kan ændres):

1. Valg af dirigent og referent

2. Formandens beretning

3. Kassererens beretning

4. Godkendelse af regnskab

5. Information fra SysOp

6. Forslag fra medlemme (ikke modtaget nogen endnu)

7. Valg af ny bestyrelse (stil op!)

8. Valg af revisor

9. Eventuelt

10. Løst og fast (afslutning)

Formanden



DisARMing

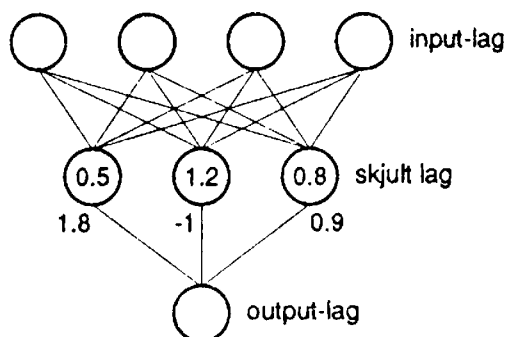
Om neurale netværk, C, assembler og bitfileri

Efter en mindre pause vender artiklerne om maskinkode til Archimedes fryygteligt tilbage. Denne artikel vil i korte træk omhandle følgende:

- 1) Princippene bag et back-propagation neuralt netværk. En implementation i C af et netværks-program.
- 2) Brugen af et PD profil-program, der kan analysere tidsforbruget i dine c-programmer.
- 3) Hvordan man kan linke assemblerprogrammer sammen med programmer skrevet i C. Hertil benyttes Acorns objektassembler.
- 4) Eksempler på optimering af netværksprogrammet med maskinkode, herunder rutiner til flydende-kommatals multiplikation og addition.

Neurale netværk

Allerførst vil jeg anbefale at finde noget litteratur om neurale netværk, hvis man er interesseret. Følgende bliver en lyngennemgang. Der findes flere forskellige typer af netværk, men ideen er i store træk den samme for dem alle sammen. Vi vil her kun beskæftige os med de såkaldte FEEDFORWARD, BACKPROPAGATE-ERROR netværk. Et netværk kan fremstilles som en mængde knudepunkter forbundet med hinanden gennem forbindelser, også kaldet vægte.



Det øverste og nederste knudelag kaldes input-henholdsvis outputlaget, alle andre kaldes skjulte lag. Kun meget simple opgaver kan løses af net uden et eller flere skjulte lag. Det er lettest at tænke på et net som en "sort kasse". Man giver netværket nogle inddata, og ud kommer et svar, som er beregnet ved at de pågældende inddata "fødes fremad" i nettet. Dette betyder at en knude i laget efter inputlaget får en talværdi, der fremkommer ved for alle forbindelser (vægte) til knuden at gange denne vægt med den knudes værdi, hvorfra vægten kommer, og summere alle disse værdier. Videreførslen gennem nettet sker på samme måde, nu bruges blot de just beregnede knudeværdier. I eksemplet vil knude nr. 2 i lag nr. 2 få værdien 0.xxxx osv. Af årsager vi ikke vil gå ind på, bruges her en aktiveringsfunktion. Dvs. de talværdier knuder tildeles, udsættes for en funktion inden de "gemmes". Det svarer i neurologiske termer til, at en neuron ikke bare sender det samlede signal den modtager videre, eller at den har en form for graderet tærskel. En sådan funktion kaldes aktiveringsfunktionen og kunne f.eks være

$$A(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$

også kaldet stepfunktionen. Mere almindelig og anvendelig er funktionen $TANH(x)$ også kaldet hyperbolsk tangens, der er en slags udjævning af stepfunktionen. Findes i c-biblioteket *math*.

Et færdigt netværk bruges ved at der sættes værdier på input-laget, det kunne f.eks være 64 stk. 0 eller 1 som i mit eksempel. Derefter fødes der fremad, og nettet giver et svar. Det eksempel, jeg refererer til, går ud på, at nettet skal genkende et bitmap af 5 tegn fra systemfonten. Svarene skal være f.eks 0.0 for tegn1, 0.1 for tegn2 osv. Inden indlæringen begynder, tildeles alle vægte tilfældige værdier, i mit eksempel mellem 0.5 og -0.5. Når der nu fødes, vil man få nogle helt

ubrugelige svar for alle eksempler, medmindre et usandsynligt tilfælde indtræffer. Prøv evt. at lege lidt med et selvkonstrueret eksempel, helst et lidt mindre net...

Nu kommer backprop-strategien ind i billedet. På et givet tidspunkt er netværket i en tilstand, der bestemmes af vægtenes værdier. For at træne netværket sammenligner man nu det egentlige output med det ønskede for hver knude i outputlaget. Kort sagt sender man nu de fundne fejl tilbage, baglæns op gennem nettet, og justerer vægten derefter. Der findes flere strategier, men den vi vil bruge er følgende: For alle eksempler gemmes de vægtændringer man ville have foretaget; de summeres gennem eksemplerne - i øvrigt i en variabel kaldet DDW. Når alle eksempler har været gennem nettet ændres vægten W ved at den tidligere vægtændring, kaldet DW, ganges med en slags fortidsfaktor og den nye vægtændring DDW med en nutidsfaktor. Det er en slags træghed man sætter ind i systemet her for at det ikke skal stå og svinge helt vildt. I mit eksempel er fortidsfaktoren, IMPULS, lig 0.9 og nutidsfaktoren, EPS, lig 0.1. Du opfordres her som mange andre steder til at eksperimentere, som du finder det passende.

Hvis nettet ikke svarer rigtigt på alle indlæringsseksempler, ændres vægten som beskrevet ovenfor og man prøver nu igen og igen og igen... Heldigvis kan et computerprogram gøre alt dette for en, evt. special-hardware. Hvis man har valgt en net-topologi, der kan løse problemet, og en fornuftig organisering af ind- og uddataværdier osv., osv. vil nettet til sidst kunne svare rigtigt på alle eksempler. Det foreliggende eksempel vil med de første fem tal-bitmaps som indlæringsseksempler have lært lektien på 40 gennemkørsler, hvis der ikke fuskes med randomgeneratoren, når vægten tildes deres startværdier. Nogle gange vil man løbe ud i noget snavs, som regel af de lige nævnte grunde, men forbavsende ofte virker det. Hvorfor er et meget interessant spørgsmål; man kan lege med et lillebitte eksempel og overbevise sig selv om, at det vil virke, men hvad der egentlig sker i et kæmpenet med mange skjulte lag er lidt dunkelt.

For den der vil læse mere om neurale netværk, så har Gyldendal lige udgivet:

Vidunderets lærlinge, introduktion til neurale netværk af William F. Allman

202 sider, kr. 198

Jeg vil nu overlade resten til kommentarerne i c-programmet NET. Jeg har udeladt nogle detaljer i backprop-strategien - men denne artikel skulle heller ikke så gerne gå hen og blive søvndyssende. Jeg henviser til litteraturen sidst i artiklen.

Optimering

Denne type opgaver er meget tunge beregningsmæssigt og derfor et godt eksempel på, hvorledes det kan betale sig at optimere, bare lidt, men på de rigtige steder. Det allervigtigste er naturligvis at vælge nogle gode algoritmer. Hvis det ikke er kendt allerede er bubblesort-algoritmen udmærket; den virker altid, den er let at programmere osv. Den er bare umåleligt ineffektiv i forhold til visse andre algoritmer, når datamængderne bliver store. Om så bubblesort bliver skrevet i maskinkode på en Cray vil den være trættende sløv i forhold til f.eks. heapsort i BASIC på en PC, hvis bare datamængderne bliver store nok (meget..)

Jeg vil ikke garantere for det vedlagte programs algoritmer, men mener dog at have fået analyseret ned til basisoperationer, der så er organiseret efter bedste evne. Nu kan man så teste sit program ved myriader af OS_ReadMonotonicTime i en eller anden afskygning, altså at spørge på tiden før og efter kald af funktioner osv. Det kan selvfølgelig blive meget præcise mål, man får på denne måde, men det kan også være noget besværligt.

Meget nemmere er det at bruge en profiler. På Quercus-bbs ligger en c-profiler, et meget net lille stykke kode, der dog gør noget ganske simpelt. Hvis man ikke slår det fra (vi taler nu om Acorns c-compiler) vil der i den oversatte kode ligge funktionsnavne. Dette benytter profileren sig af således: Man inkluderer profileren og kalder allerøverst i main() `exprofile_init(fil)`. Den gennemløber koden og noterer sig funktionsnavne men ligeså vigtigt adresser. Derefter sættes en interruptrutine op, der bliver kaldt 100 gange i sekundet. Når profileren således får kommandoen bruges returadresseinformation på snedig vis til at beregne, hvorfra programmet blev afbrudt. Hvis det er i en kendt rutine tælles dennes variabel en op. Nu bruges der gængs matematik til at udregne tidsprocenter og dette ligger rede til inspektion efter programudførelsen, uden at man behøver at foretage sig nogetsomhelst andet. Simple and Smart. Prøv det selv, men her følger et lille uddrag fra en netkørsel. Alle ikke medtagne funktioner havde count 0.


nr	address	function name	percentage	count
2	8148:	layer_init	: 0.26,	12
3	8218:	layer_feedforward	: 20.11,	912
4	82dc:	layer_backprop	: 56.76,	2574
5	8424:	layer_weighteval	: 16.01,	726
8	8590:	net_feedforward	: 0.71,	32
9	861c:	net_backprop	: 0.04,	2
11	8700:	net_test	: 0.04,	2
31	a8f0:	greaterthan	: 6.06,	275
			100.00,	4535
percentage of time spent in fpemulator				
32	183a584:	fpemulator :	96.30,	4367

Normalt vil man kunne forudsige et programs tidsforbrugsfordeling nogenlunde uden målinger af nogen art, men det er nu alligevel meget interessant at se de mere præcise tal. For det første er det lidt overvældende, så meget tid der bruges i floating-point emulatoren, mens det ikke er så mærkeligt at de tre lag-rutiner fylder næsten det hele. De sidste laver også det meste arbejde, de basisoperationer jeg omtalte før er jo noget med at loadere og store værdier og lave beregninger på dem ind i mellem.

De kunne se ud som om man skulle skrive rutinerne 3-5 i assembler, men mere ville afgjort kunne vindes ved at skrive specielle gange og additions-rutiner og bruge dem i stedet for emulatorens. Vi bruger i forvejen den mindste præcision på 32 bits, så der er ikke noget at hente. Se i øvrigt en af de tidligere artikler for detaljer om FPE og om flydende tal.

For at gøre dette skal man være i besiddelse af Acorns objektassembler eller selv producere det nødvendige format ud fra output fra en anden assembler. Nu skal man gøre nogle ganske få ting anderledes end med BASIC's assembler, der skal en anden header på, for at linkerens kan finde rundt i det hele og enkelte andre ting. Desuden skal vi vide, hvordan man bliver kaldt fra c, dvs.

hvilke registre indeholder argumenter, hvilket indhold man må destruere osv. Dette er meget let; hvis der er op til fire argumenter ankommer de med en BL instruktion og findes i R0 og opefter. Stakpointeren er som sædvanligt R13 og skal tælles ned inden man pusher. Du skal i øvrigt blot erklære dine assemblerrutiner som eksterne, se prg. Som oftest er der ikke så meget pjat i c. Det er også rart med c, at man kan se sine højniveau algoritmer afspejlet med knusende systematik, især omkring pointere, brug dem tit og ofte. Du kan nemlig få c-compileren til at generere assembler kode og undersøge denne, det er absolut rimelige resultater den producerer, sådan da. Man kan jo altid pine lidt, som vi snart vil gøre - men håb ikke på mirakler. De fås bedre ved at finde en genial algoritme.

Udover net der er skrevet i c, følger der assembler for de tre tunge drenge fra før og desuden floating-point multiplikation og addition. Du er nødt til at lave nogle arrays, *lex*, hvor hver indgang har 65 reals, 64 for bitmappen og et for det ønskede output. Jeg har valgt 1.0 for satte bits og 0.0 ellers. Egentlig så det hele pænere ud med FPE-instruktioner men det er ikke så mærkeligt, da man her får 8 nye registre forærende (software endnu). Isoleret set skulle *mul* og *add* være ca. 8 og 15 gange henholdsvis hurtigere end 

emulatorens funktioner, og det vil selvfølgelig give noget fart. Koden er sikkert ikke noget særligt, så det virker lidt mærkeligt. Se den tidligere artikel for detaljer om emulatorens interne formater. De har større præcision internt og kan sikkert håndtere alle situationer, hvad jeg ikke vil påstå mine kan, men alligevel... Desværre

skal der laves andet end gange og plus, og compileren er meget sværere at tackle. Man kan som sagt ikke hekse. Der vil dog altid være forbedringer at finde og de er særligt velkomne når opgaven er stor. Der følger et profiler snapshot med hele pivetøjet slået til efter samme princip som før.

nr	address	function name	percentage	count
2	8148:	layer_init	: 1.04,	13
8	8590:	net_feedforward	: 2.55,	32
9	8620:	net_backprop	: 0.16,	2
11	8708:	net_test	: 0.24,	3
13	8804:	list_out	: 0.08,	1
16	89b4:	ffeed	: 5.18,	65
17	8a40:	backp	: 8.36,	105
18	8b1c:	weval	: 1.43,	18
19	8b98:	fmul	: 36.78,	462
20	8c2c:	fadd	: 22.21,	279
31	a8f8:	greaterthan	: 21.97,	276
			100.00,	1256
percentage of time spent in fpemulator				
32	183a584:	fpemulator	: 24.12,	303

Man kan nu begynde at se basisoperationerne tone frem. Samlet er programmet selvfølgelig ikke 11-12 gange hurtigere end c-programmet, men derimod ca. 4 gange hurtigere, som det ses. Desuden fylder FPE stadig godt i landskabet, selvom dens indflydelse i de tre tungeste funktioner er fjernet næsten totalt, utroligt. Der må alligevel være mere overhead, end man tror. Jeg har forsøgt at overholde FPE-formatet også for de helt patologiske tilfælde, så hvis nogen får fornemmelser fra emulatoren vil jeg gerne høre om det. Man kunne lave et privat format, men det ville give problemer for de andre c-funktioner, der jo stadig har FPE-instruktioner. Desuden kaldes c-funktionen TANH fra *feedf*. En

optimering her ville være en slags opslagsfunktion til *tanh* istedet for FPE-beregning; hvis nogen laver sådan en til næste gang får han helt sikkert lov at afbryde mine udgydelser. Bare skriv. Vi vil dog kun bringe programmer, der er kommenterede. Er der ikke også nogen wimp-eksperter, der laver en front-end til et neuralt Quercus-netværksprogram? En anden idé - en dansk version af floatingpointemulatoren, mens vi venter på en hardwareversion.. Jeg håber kommentarene i programmerne kan klare resten, men interesse af enhver art modtages med kyshånd. Programmerne kan findes paa boardet som "neural" under *applications*.

M.V.H. Hans Otto Lunde 6544 

C-programmet "net" til DisARMing:

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <float.h>
#include <limits.h>
#include "bbc.h"

#define EXPRO      1
#include          "exprofle.h"    /* inkluder profilerkode      */
#define TRUE      1
#define FALSE     0
#define PF        1              /* l=forwardfeed i assembler */
#define BP        1              /* l=netbackprop i assembler */
#define WE        1              /* l=net_weighteval i assembler */
#define LAYERS    4              /* antal lag                   */
#define NO_EX     5              /* antal indl ringseksempler  */
#define IN        64             /* antal inputkuder           */
#define OUT        1             /* antal outputkuder          */
#define TOLERANCE 0.05           /* fejltolerance               */
#define IMPULSE   0.9            /* fortidens v gtning          */
#define EPS       (1.0 - IMPULSE) /* nutidens v gtning           */
#define real      float         /* kommatal, 32-bits          */

typedef struct {                  /* record for et lag          */
    int in;                       /* antal inputs               */
    int out;                       /* antal outputs              */
    real *w;                       /* pointer t. v gte            */
    real *dw;                       /* pointer t. v gt ndringer    */
    real *ddw;                       /* pointer t. gemte v gt ndr.  */
    real *data;                       /* pointer t. outputdata      */
} layer;

int topologi[LAYERS] = { 64,16,4,1 }; /* netv rkets topologi      */
layer net[LAYERS];
extern real lex[];                  /* eksempeldata              */
extern void ffeed(real *,layer *); /* machine-code routines..  */
extern void backp(real *,layer *);
extern void weval(layer *);

void layer_alloc(layer *l,int in,int out) /* alloker plads til lag */
{ int vectorsize = (in+1)*out;
  l->out = out;
  l->data = (real *) malloc(out*sizeof(real));
  if (l->in = in)
  { l->w = (real *) malloc(3*vectorsize*sizeof(real));
    l->dw = l->w+vectorsize;
    l->ddw = l->dw+vectorsize; } }

void layer_init(layer *l)           /* s t tilf ldige v gte m.m.*/
{ int n;
  for (n = 0; n < l->out; n++) *(l->data+n) = -1.0;
  if (l->in)
    for (n = 0; n < l->out*(l->in+1); n++)
      { *(l->w+n) = ((real) rand()/RAND_MAX) - 0.5;

```

```

        *(l->dw+n) = *(l->ddw+n) = 0.0; } }

void layer_feedforward(real *inp, layer *l)
{ int i,o; real tmp,*p;
  for (o = 0; o < l->out; o++)
    { tmp = - *(l->w + o); /* bias -1 */
      for (p = l->w+o+l->out, i = 0; i < l->in; i++, p += l->out)
        tmp += *(inp+i) * *(p);
      *(l->data+o) = tanh(tmp); } }

void layer_backprop(real *inp, layer *l)
{ int i,o; real *p,d;
  for (o = 0; o < l->out; o++) /* foralle o */
    { *(l->ddw+o) -= *(l->data+o);
      for (p = l->ddw+o+l->out, i = 0; i < l->in; i++, p += l->out) /* og i */
        *(p) += *(l->data+o) * *(inp+i); }
  for (p = l->w+l->out, i = 0; i < l->in; i++) /* foralle i */
    { for (d = 0.0, o = 0; o < l->out; o++, p++) /* foralle o */
      d += *(l->data+o) * *(p); /* summer W*OUTPUT i d */
      *(inp+i) = d*(1 - *(inp+i)* *(inp+i)); } } /* sæt output i lag-l */

void layer_weighteval(layer *l)
{ int idx; real tmp;
  for (idx = 0; idx < l->out*(l->in+1); idx++)
    { tmp = (*(l->dw+idx)*IMPULSE) + (*(l->ddw+idx)*EPS);
      *(l->w+idx) += tmp; /* lav ny vægt */
      *(l->dw+idx) = tmp; /* lav ny DW */
      *(l->ddw+idx) = 0.0; } } /* reset DDW */

void net_alloc() /* alloker de nødvendige lag */
{ int n;
  layer_alloc(&net[0],0,topologi[0]);
  for (n = 1; n < LAYERS; n++)
    layer_alloc(&net[n],topologi[n-1],topologi[n]); }

void net_init() /* initialiser alle lag */
{ int n;
  for (n = 0; n < LAYERS; n++) layer_init(&net[n]); }

void net_feedforward(real *ex) /* feedforward gennem net */
{ int n;
  for (n = 0; n < net[0].out; n++) *(net[0].data+n) = *(ex+n);
  for (n = 0; n < LAYERS - 1; n++)
    if (PF == 1) ffeed(net[n].data,&net[n+1]);
    else layer_feedforward(net[n].data,&net[n+1]); }

void net_backprop(real *ex) /* backprop gennem net */
{ int n;
  for (n = 0; n < net[LAYERS-1].out; n++)
    *(net[LAYERS-1].data+n) = *(ex+n) - *(net[LAYERS-1].data+n);
  for (n = LAYERS-1; n > 0; n--)
    if (BP == 1) backp(net[n-1].data,&net[n]); /* assembler.. */
    else layer_backprop(net[n-1].data,&net[n]); } /* eller c */

void net_weighteval() /* nye vægte i net */
{ int l;
  for (l = 1; l < LAYERS; l++)
    if (WE == 1) weval(&net[l]); else layer_weighteval(&net[l]); }

```

```

int net_test(real *lex)                                /* test output */
{ int n; real *p = net[LAYERS-1].data;
  for (n = 0; n < OUT; n++)
    if (fabs(*(p+n) - *(lex+n)) > TOLERANCE) return(FALSE);
  return(TRUE); }

void list_out(real *ex)                                /* udskriv output */
{ int n;
  for (n = 0; n < OUT; n++) printf("%9f ",*(ex+n)); }

int learn_lex()
{ real *ex; int err,i; i = err = 0;
  bbc_vduq(bbc_MoveText,0,0);
  for (ex = lex; ex < lex+NO_EX*(IN+OUT); ex += IN+OUT,i++)
    { net_feedforward(ex);
      list_out(net[LAYERS-1].data);
      if (!net_test(ex+IN)) err++;
      net_backprop(ex+IN);
      printf(" <- %c\n", '0'+i); }
  if (err) net_weighteval();
  return(err); }

int main()
{ int err,lesson; err = lesson = 0;
  exprofile_init("profile"); /* profileroutput i "profile" */
  bbc_vduq(bbc_ClearText);
  net_alloc(); net_init();
  while (err = learn_lex())
    printf("\nlesson %4d total errors: %4d\n",++lesson,err);
  return(0); }

```

Assembler-rutiner til DisARMing:

r0	RN	0	;r0-r3 indeholder argumenter, medmindre de
r1	RN	1	;der er for mange. Kan frit benyttes
r2	RN	2	
r3	RN	3	
r4	RN	4	;r4-r11's indhold skal bevares
r5	RN	5	
r6	RN	6	
r7	RN	7	
r8	RN	8	
r9	RN	9	
r10	RN	10	
r11	RN	11	



```

r12 RN 12
sp RN 13 ;sp=fuld, nedadvoksende stak, skal bevares
lr RN 14 ;retur-adressen er i lr
pc RN 15
f0 FN 0 ;vi undgår ikke helt PPEmulatoren

```

```
AREA |C$$code|,CODE,READONLY ;runer til objasm og linkerens
```

```

DCB "fmul",&00
ALIGN
DCD &ff000008
EXPORT fmul ;multiplikation (r0,r1) --> r0

fmul STMDB sp!,{r2-r6} ;vi får brug for r2-r6, stak dem
MOV r5,r0,LSR#31
SUB r6,r5,r1,LSR#31 ;r6=fortegnssdifference
MOV r4,#&80000000
MOV r5,#255
ANDS r2,r5,r0,LSR#23 ;r2=expl
ORRNE r0,r4,r0,LSL#8 ;hvis expl<0, dan mantissel
MOVEQ r3,#0 ;ellers sikr exponentresultat
ANDNES r3,r5,r1,LSR#23 ;r3=exp2
ORRNE r1,r4,r1,LSL#8 ;som ovenfor..
MOVEQ r2,#0
MOVEQ r0,#0 ;z-flag sat her -> res = 0
ADDNE r2,r2,r3 ;adder exponenterne, or normaliser
SUBNE r2,r2,#127 ;r2=midlertidigt exponent-resultat
mul164 MOV r5,r0,LSR#16 ;r5=high1, der følger 64bits heltalsmul
MOV r4,r1,LSR#16 ;r4=high2
BIC r0,r0,r5,LSL#16 ;r0=low1
BIC r1,r1,r4,LSL#16 ;r1=low2
MUL r3,r0,r1 ;r3=low1*low2
MUL r1,r5,r1 ;r1=high1*low2
MUL r0,r4,r0 ;r0=high2*low1
MUL r4,r5,r4 ;r4=high1*high2
ADDS r0,r1,r0 ;r0=high1*low2 + high2*low1 + carry
ADDCS r4,r4,#&10000 ;r4+=carry in bit 16
ADDS r3,r3,r0,LSL#16 ;r3=lowdel af resultat
ADC r4,r4,r0,LSR#16 ;r4=highdel.....
nmul TST r4,#&80000000 ;hvis bit31 sat
MOVNE r4,r4,LSL#1 ;skift underforstået bit ud og
ADDNE r2,r2,#1 ;læg 1 til exp
MOVEQ r4,r4,LSL#2 ;ellers skift 0 + underforstået bit ud
MOVS r0,r4,LSR#9 ;mantisse på plads i r0, carry=r4,bit9
ADDCSS r0,r0,#1 ;hvis sidste "tabte" bit sat, juster r0
CMP r2,#255 ;test maximumtilfælde
MOVGE r2,#254
CMP r6,#0
ADDNE r2,r2,#256 ;sæt evt. minusfortegn foran exp
ORR r0,r0,r2,LSL#23 ;dan resultat
LDMIA sp!,{r2-r6} ;genopret arbejdsregistre
MOVS pc,lr ;returner til sidste omgivelser m. flag

DCB "fadd",&00
ALIGN
DCD &ff000008
EXPORT fadd ;addition (r0+r1) --> r0

```



```

EXPORT fsub ;subtraktion (r0-r1) --> r0

fsub EOR r1,r1,#&80000000 ;=fadd(arg1,-arg2), snyderutine, ikke?
fadd STMDB sp!,{r2-r6,lr} ;r0-r5 er nok her..
MOV r5,#255
AND r2,r5,r0,LSR#23 ;r2=expl
AND r3,r5,r1,LSR#23 ;r1=exp2
MOV r4,r0,LSR#31 ;r4=fortegn1
MOV r5,r1,LSR#31 ;r5=fortegn2
CMP r5,r4 ;test fortegn
BNE sub ;sub hvis ikke ens, ellers adder
add MOV r0,r0,LSL#9 ;r0=mantissel,venstrestillet
MOV r1,r1,LSL#9 ;r1=mantisse, .....
SUBS r4,r2,r3 ;r3=exponentforskell
BLT dda ;exp2 er størst - hop
MOVEQ r0,r0,LSR#1 ;exponenter ens, højreskift
MOVEQ r1,r1,LSR#1 ;begge mantisser
ADDEQ r2,r2,#1 ;juster exp en op
MOVNE r1,r1,LSR#1 ;expl størst, skift og sæt impl. bit
ORRNE r1,r1,#&80000000 ;for arg2
SUBNE r4,r4,#1 ;og tæl exponentdifference en ned
ADDS r0,r0,r1,LSR r4 ;læg mantisserne sammen, justeret
ADDCS r2,r2,#1 ;hvis carry, tæl exp en op
MOVCS r0,r0,LSR#1 ;og juster mantisse
MOVS r0,r0,LSR#9 ;mantisse på plads, test den -1 bit
ADDCS r0,r0,#1 ;rund opad, hvis den var sat
CMP r2,#255
MOVGE r2,#254
ORR r2,r2,r5,LSL#8 ;da fortegn ens
ORR r0,r0,r2,LSL#23 ;skift exponent på plads
dda SUB r2,r3,r2 ;r2=exponentforskell, positiv
MOVS r0,r0,LSR#1 ;skift den mindste mantisse og
ORR r0,r0,#&80000000 ;sæt dens implicitte bit
SUB r2,r2,#1 ;tæl expforskell en ned
ADDS r1,r1,r0,LSR r2 ;foretag additionen
ADDCS r3,r3,#1 ;hvis carry, expresultat++
MOVCS r1,r1,LSR#1 ;og skift
MOVS r1,r1,LSR#9 ;dan resultat
ADDCS r1,r1,#1 ;juster opad
CMP r3,#255 ;test for max
MOVGE r3,#254
ORR r3,r3,r5,LSL#8 ;sæt fortegn
ORR r0,r1,r3,LSL#23 ;exponent på plads
LDMIA sp!,{r2-r6,pc}

sub MOV r6,#&80000000
ORR r0,r6,r0,LSL#8 ;man1 i r0, sæt implicit bit
ORR r1,r6,r1,LSL#8 ;man2 i r1...
SUBS r6,r2,r3 ;r6=expforskell
CMPEQ r0,r1 ;hvis expl=exp2, test mantisser
BLT bus ;hvis arg1<arg2, vend som i add
MOV r3,r1,LSR r6
CMP r0,#&80000000
BICEQ r3,r3,#255
SUBS r0,r0,r3 ;resman=man1 - man2
MOVEQ r2,#0
norm MOVS r0,r0,LSL#1 ;ellers venstreskift mantisseresultat

```

```

        BEQ      nonor
        SUBCC   r2,r2,#1          ;og tæl exponent ned
        BCC     norm              ;indtil en bit dukker op
nonor   MOVCS   r0,r0,LSR#9       ;dan resultat
        ADDCS   r0,r0,#1         ;rund evt. op
        CMP     r2,#255
        MOVGE   r2,#254
        ORR     r2,r2,r4,LSL#8    ;skift exp på plads
        ORR     r0,r0,r2,LSL#23   ;skift fortegn+exp på plads
        LDMIA   sp!,{r2-r6,pc}~
bus     SUB     r6,r3,r2          ;r6=expforskel
        MOV     r2,r0,LSR r6
        CMP     r1,#&80000000
        BICEQ   r2,r2,#255
        SUBS    r0,r1,r2          ;res=man2-man1
        MOVEQ   r3,#0
mron    MOVCS   r0,r0,LSL#1       ;ellers normaliser som ovenfor
        BEQ     r0non
        SUBCC   r3,r3,#1         ;hvis 0 normaliser ikke
        BCC     mron             ;skift til venstre
ronon   MOVCS   r0,r0,LSR#9       ;mantissee på plads
        ADDCS   r0,r0,#1         ;rund evt. op
        CMP     r3,#255
        MOVGE   r3,#254
        ORR     r3,r3,r5,LSL#8    ;fortegn og exponent
        ORR     r0,r0,r3,LSL#23   ;exponent på plads
        LDMIA   sp!,{r2-r6,pc}~

        DCB     "ffeed",&00      ;functionnavnet i koden, bruges
        ALIGN   ;bl.a af profileren
        DCD     &fff00008        ;exporter navnet, så man kan kalde fra c..
        IMPORT  tanh
        EXPORT  ffeed

ffeed   STMDB   sp!,{r4-r11,lr}   ;r0=*inp,r1=*layer
        MOV     r6,r0             ;r6=*inp
        LDMIA   r1,{r0,r2,r3}     ;r2=out,r3=*w
        LDR     r5,[r1,#20]       ;r5=*data
        MOV     r8,r0
        MOV     r10,r2
        MOV     r7,r0             ;r7=in
        MOV     r4,r10           ;r4=out
for_o   ADD     r8,r3,r10,LSL#2    ;r0=w+o+out=p
        LDR     r2,[r3],#4        ;*(w+o)
        EOR     r2,r2,#&80000000
        MOV     r11,r7           ;lr=in-count
        MOV     r9,r6            ;r1=inp
for_i   LDR     r0,[r9],#4
        LDR     r1,[r8]
        BL     fmul
        MOV     r1,r2
        BL     fadd
        MOV     r2,r0
        ADD     r8,r8,r10,LSL#2    ;p+=out
        SUBS    r11,r11,#1
        BNE     for_i
        STR     r2,[sp,#-4]!      ;gem tal som single-prec
        LDPS   f0,[sp],#4        ;ind i emulatoren

```

```

STFD    f0,{sp,#-8}!           ;gem som double-prec
LDMPD   sp!,{r0,r1}           ;kald c's tanh..
BL      tanh
STFS    f0,{r5},#4            ;inc out-pointer
SUBS    r4,r4,#1
BNE     for_o
LDMIA   sp!,{r4-r11,pc}^

DCB     "backp",&00
ALIGN
DCD     &ff000008
EXPORT  backp

backp   STMDB sp!,{r4-r12,lr}   ;r0=inp,r1=*layer
        LDMIA r1!,{r5}         ;r5=in
        LDMIA r1!,{r3-r4}      ;r3=out,r4=w
        LDMIB r1!,{r10,r11}    ;r10=ddw,r11=data
        MOV    r9,r0           ;r9=inp
        MOV    r7,#0           ;r7=0
o1      ADD    r6,r10,r7,LSL#2   ;r6=p=ddw+o
        LDR    r0,{r6}
        ADD    r12,r11,r7,LSL#2 ;r12=data+o
        LDR    r1,{r12}
        MOV    r2,r1           ;r2=*(data+o)
        BL     fsub
        STR    r0,{r6}
        MOV    r12,r9          ;r12=tmp *inp
        MOV    r8,r5           ;r8=in
        ADD    r6,r6,r3,LSL#2   ;p=ddw+o+out
i1      LDR    r1,{r12},#4
        MOV    r0,r2
        BL     fmul
        LDR    r1,{r6}
        BL     fadd
        STR    r0,{r6}
        ADD    r6,r6,r3,LSL#2   ;p+=out
        SUBS   r8,r8,#1        ;more inp's..?
        BNE   i1
        ADD    r7,r7,#1        ;o++
        CMP    r3,r7
        BGT   o1

        ADD    r10,r4,r3,LSL#2  ;p=w+out
i2      MOV    r2,#0           ;floating point zero
        MOV    r6,r3           ;r6=out
        MOV    r4,r11          ;r4=now temporary *data
o2      LDR    r0,{r4},#4
        LDR    r1,{r10},#4
        BL     fmul
        MOV    r1,r2
        BL     fadd
        MOV    r2,r0
        SUBS   r6,r6,#1
        BNE   o2
        LDR    r0,{r9}
        MOV    r1,r0
        BL     fmul
        MOV    r1,r0

```

```

MOV    r0,#&3f800000      ;floating point 1.0..
BL     fsub
MOV    r1,r2
BL     fmul
STR    r0,{r9},#4
SUBS   r5,r5,#1           ;any more inputs..?
BNE    i2
LDMIA  sp!,{r4-r12,pc}^

DCB    "weval",&00
ALIGN
DCD    &fff000008
EXPORT weval

weval  STMDB   sp!,{r4-r11,lr} ;r0=&lager ved indgang
LDMIA  r0,{r3-r7}           ;r3=ind,r4=ud,r5=*w,r6=*dw,r7=*ddw
MLA    r4,r3,r4,r4         ;r4=max vægte (in+1)*out
MOV    r3,#0               ;r3=hurtig 0.0
LDR    r8,[pc,#imp--8]     ;r7=0.9
LDR    r9,[pc,#eps--8]    ;r8=0.1
loop   LDR    r10,[r5]      ;r10=(1->w+idx)
LDR    r1,[r6]             ;r1=(1->dw+idx)
LDR    r11,[r7]           ;r11=(1->ddw+idx)
MOV    r0,r8
BL     fmul                ;r0=dw*impuls
MOV    r2,r0
MOV    r0,r9
MOV    r1,r11
BL     fmul                ;r0=ddw*eps
MOV    r1,r2
BL     fadd                ;r0=dw*impuls+ddw*eps
MOV    r2,r0
MOV    r1,r10
BL     fadd                ;r2=tmp
BL     fadd                ;r1=(1->w+idx)
BL     fadd                ;r0=w+=tmp
STR    r0,[r5],#4         ;gem ny vægt
STR    r2,[r6],#4         ;gem ny dw
STR    r3,[r7],#4         ;ddw=0.0
SUBS   r4,r4,#1
BGT    loop
LDMIA  sp!,{r4-r11,pc}^

imp    DCFS    0.9
eps    DCFS    0.1

END

```

Adresseliste

Nye indmeldelser m.v.

NYE MEDLEMMER:

Indmeldt i perioden 16/02/91 - 20/07/91

6507

Raymond Cox
Biokemisk institut
Campusvej 55
5230 Odense M

6508

Sergent
Aage Jensen
Viborgvej 32
7470 Karup J
Tlf.: 97 10 25 82

6509

André Grafik
v/ Michael Rothman
Gråpilevej 6
2880 Bagsværd
Tlf.: 42 98 40 46

6555

Kjeld Mejndor
Karetmagervej 27
8900 Randers
Tlf.: 86 42 78 42

6565

Erik Axel
Vester Søgade 54, st.th.
1601 København V
Tlf.: 33 14 37 24

6572

Erik Vind Nielsen
Grejsdalsvej 176
7100 Vejle
Tlf.: 75 82 23 77

UDMELDLSER:

Følgende numre er slettet af medlemslisten:

6524 6539 6561 6594 6595

ADRESSEÆNDRINGER:

6511

Elektronikmekaniker
Ole Nebel Nielsen
Lindegade 5B
6823 Ansager

6523

Thomas Midtbøll
Amager Boulevard 125, 1.th.
2300 København S

6563

Peter Holst Andersen
Ved Vænget 5, st.th.
2100 København Ø
Tlf.: 31 38 33 48

6623

Mads Meisner Jensen
Valdemarsgade 6, st.tv.
9000 Aalborg

6639

Réne Rasmussen
Rørsangervej 19, 1.th.
2400 København NV

6640

Morten Rethmeier
Gråskærgade 39 st.th.
6100 haderslev

6625

Glenn Møller-Holst
Rektorparken 1, vær. 520
2450 København SV

6534

Stefan Nilsson
Nybrovej 356, P17
2800 Lyngby

Klubside

Månedens tilbud til klubbens medlemmer Støt klubben - bestil nu!

Kontingent

Abonnement på Egebladet (6 blade) + medlemskab i et år: kr. 175,00

Tidligere numre af Egebladet. nr. 1-7 pr. stk: kr. 12,00, nr. 8-16 pr. stk: kr. 20,00. Hele årgang 1988 (1-4): kr. 40,00. Årgang 1989 (5-10): kr. 80,00. Årgang 1990-91 (11-16): kr. 80,00.

Ekstra badges (medlemskab inkluderer et gratis badge): kr. 10,00 pr. stk.

Disketter

Klubdiskette nr. 1 (BBC og Master). Listninger fra blad nr. 1-3, samt biorytmeprogram, raytracer og badgeprogram. kr. 20,00

Klubdiskette nr. 2 (BBC og Master). Listninger fra blad nr. 4-7, samt væveprogram, vinderbadget og vinduedemo, kr. 20,00

Klubdiskette nr. 3 (BBC og Master). Listninger fra blad nr. 8-16, samt programmer til artiklen om sideways RAM (blad nr. 16)

Klubdiskette nr. 1 (Archimedes og A3000). Listninger fra bladene plus programmer fremstillet af vores medlemmer, Kr. 25,00

Bemærk: Ovenstående priser er alle inklusive porto.

Særlige disketter:

Medlemsregister, vedtægter. Send frankeret svarkuvert og formatteret diskette.

Gallery-disks (til Archie og A3000); 5 disketter fyldt med IFF/GIF billeder, kr. 100,00

Indeks (BBC og Master serien). Har du problemer med at huske hvilke disketter der indeholder bestemte spil, så er dette programmet for DIG. Det eneste der kræves er at dine disketter indeholder en menu. Indeks går ind og læser alle data i menuen og gemmer disse til f.eks. udprintning. Vi sælger dette eminente program til den latterlige lave pris af kun kr. 25,00.

Programmer på tape (BBC og Master).

Skriv efter en komplet liste. Listen er også at finde på vores BBS.

1 bånd: kr. 10,00, 4 bånd: kr. 30,00, 10 bånd: kr. 75,00. (Vi har også programmer på disk fra kr. 25,00 og opad). (BEMÆRK; VI HAR SÆNKET PRISERNE).

Ved bestillinger på kr. 50,00 og derover betales der ikke forsendelsesporto.

Opgiv navn og adresse samt medlemsnummer.

Martin J. Søndergaard

Årestrupvej 76

7470 Karup J

Tlf. 97 10 21 95

BEMÆRK: Vi mangler stadig programmer fra medlemme til de næste klubdisketter, så få nu fundet de små (u)nyttige programmer frem, så andre kan få glæde af dem.



Programmering

...kunst eller håndværk

- en ny artikelserie om programmering på Acorn-maskiner.

Da jeg sagde ja til at redigere dette nummer (for flere måneder siden), havde jeg tænkt på at lave et tema-nummer ud af det - med vægten lagt på programmering. Trods alt er det jo nok det, de fleste Arc/BBC-brugere anvender deres maskine til(?). Det er dog ikke lykkedes mig at få dette tema-nummer lavet, primært fordi det ikke er lykkedes mig at aktivere så forfærdelig mange af programmørerne i klubben, som jeg havde troet. Lidt om programmering er der dog - se blot Hans Otto Lundes artikler.

En ny slags artikel

Da jeg altså ikke fik stof nok til at lave et decideret programmerings-nummer, fik jeg så den idé at jeg jo selv kunne producere en artikel - eller nærmere en artikelserie. Og dette er altså den første artikel i en serie. Hvor lang serien bliver, ved jeg ikke endnu. Det kommer til at afhænge af dig! Denne artikelserie bliver nemlig anderledes end de fleste andre, der har været her i bladet, idet den bliver interaktiv: Dvs. at alle I læsere derude kan være med til at påvirke indholdet af serien. Artiklen i hvert nummer vil blive produceret kort tid før deadline, og den vil medtage læsernes kommentarer og forslag (hvis altså der kommer nogle). På den måde skulle vi gerne få en levende og spændende artikelserie ud af det. På den anden side: Hvis der ikke kommer nogle forslag eller kommentarer fra læsere, vil jeg måske lade serien hvile lidt, indtil der sker noget! Så hvis du er interesseret i fortsættelsen af serien, så skriv ind!

Baggrund

Nå, men nok om snakken; lad os se at komme igang. Først vil jeg lige kort fortælle om min egen baggrund: Jeg har i mange år (siden jeg fik min Atom i starten af 80'erne) programmeret i BASIC

og 6502-assembler (samt COMAL på skolen). Dengang var det mest selvlært (eller med et fint ord: autodidakt). Det, der fascinerede mig mest ved computere var det, at man kunne få dem til at gøre noget - hvis man bare bad dem om det, ville en computer gøre hvad som helst (indenfor mulighedernes grænse naturligvis). Man kunne sågar få den til at spille (dengang var det mest kryds og bolle, tændstikspil og simple kortspil). Senere fik jeg lyst til at systematisere min viden, og startede på EDB-skole. Jeg er nu blevet EDB-assistent og er knap halvvejs igennem en datamatiker-uddannelse. Jeg må dog indrømme, at jeg er langt fra en professionel programmør, men jeg kan tilfældigvis lide at programmere (uanset om mine programmer så bliver bedre af den grund). Derfor skal man ikke være bange for at kritisere mine arbejdsmetoder; det er netop det, der er meningen med denne serie.

Programmeringssprog

Hvilke muligheder har man så på BBC/Acorn-computere for at programmere? Tja, på alle maskiner har man den indbyggede BBC BASIC fortolker, hvor der også er en inline assembler. BASIC er normalt ikke særligt velset hos programmører, dette hænger sammen med, at de fleste BASIC's (ihvertfald før i tiden) var ret ustrukturerede (f.eks. ingen muligheder for procedurer eller strukturerede konstruktioner såsom REPEAT UNTIL), og eksempelvis ofte anvendte GOTO for at styre "program-flowet". Dette har dog heldigvis ændret sig, og BBC BASIC er efterhånden gået hen og blevet et godt sprog (især med BASIC V på Archimedes/A3000). Dette er da også grunden til, at jeg i første omgang vil koncentrere mig mest om BBC BASIC. Det skal dog nævnes, at der findes en hel del andre sprog til Acorn maskiner, f.eks. Fortran, C, Pascal (se f.eks. omtalen af Cambridge Pascal i dette nummer),

Logo, COMAL, Prolog, Lisp... (har jeg glemt nogle?) - de fleste af disse i flere udgaver, både fra Acom og fra andre softwarehuse. Fælles for disse sprog er at de er såkaldte 3. generations sprog (hvor 1. generationssprog er ren maskinkode og 2. generations sprog er assembler), dvs. procedurelle - eller imperative - sprog. Sprog, hvor man helt nøjagtigt fortæller HVORDAN computeren skal gøre, istedet for at fortælle, HVAD man vil have gjort. En undtagelse fra dette er Prolog, der er et logik-programmeringssprog, og af mange betragtes som det bedste eksempel på et 5. generationssprog (det anvendes bl.a. til ekspertsystemer). En anden ting, der er fælles for de fleste af disse sprog, er at de er ret gamle - Fortran og Lisp er to af de ældste programmeringssprog overhovedet! C har over 15 år på bagen, Pascal mere end 20!

Nyere sprog

Er der da ikke sket nogen ændringer mht. programmeringssprog igennem de sidste 10-15 år? Jo, selvfølgelig er der det, men programmører må vist siges at være ret konservative - hvorfor er sprog som Fortran og COBOL ellers nogle af de mest brugte stadigvæk? Det er der naturligvis en masse forklaringer på, men lad os ikke dvæle ved det lige nu. Nogle af de mere interessante ting, der er sket indenfor computersprog i de senere år er objektorienteret programmering (eksempelvis C++), funktionel programmering (eksempelvis Miranda), logikprogrammering (eksempelvis Prolog) og 4. generations programmeringssprog (der findes en masse, der påstår at være dette, men som et eksempel, jeg selv kender, kan jeg nævne Informix 4GL).

Mangler

Hvis man checker dette med listen fra før over hvilke programmeringssprog, vi har på BBC/Acom-maskiner, vil man hurtigt finde ud af, at det vi mangler er: Objektorienterede programmeringssprog og 4. generations programmeringssprog. Disse to begreber behøver ikke at udelukke hinanden (selv om det sædvanligvis drejer sig om forskellige produkter), men er på en måde to sider af samme sag. Nøgleordet er: Abstraktion! Fra de sædvanlige 3. generations sprog (som er en abstraktion fra 2. generations sprog) skal vi kunne abstrahere; dvs. bevæge os længere væk fra maskin-niveauet, og på et

begrebsmæssigt højere plan. Det nærmeste, vi kommer f.eks. 4. generations sprog på Acom-maskiner er sådan noget som programmerbare regneark og databaser, men disse må nok siges at have en begrænset anvendelse. Objektorienterede programmeringssprog tror jeg slet ikke findes på Acom maskiner endnu (medmindre man vil programmere C++ eller Turbo Pascal 5.5 under PC-emulatoren).

BBC BASIC

For nu at komme ned i et andet plan igen, vil jeg gå tilbage til hvor vi slap, nemlig BBC BASIC. Ganske vist har vi ikke Ada eller Modula-2 på vores maskiner, men man kan sagtens programmere på et forholdsvis højt niveau (altså abstrahere) i BBC BASIC. Dette gøres primært ved hjælp af funktioner og procedurer, men også de strukturerede programkonstruktioner (REPEAT UNTIL, FOR NEXT, etc) hjælper på dette. Et af formålene med denne artikelserie, er at få belyst noget af problematikken omkring selve det at programmere. For at få så meget som muligt ud af denne artikel, vil jeg prøve at tage udgangspunkt i et konkret eksempel, og arbejde ud fra dette. Oplægget er i sin korthed følgende:

Spilsimulering

Et af de spil, der - så vidt vides - endnu ikke er lavet til Acom computere, er Dam. Der findes ganske vist enkelte udgaver, men de spiller efter de engelske regler, der er noget anderledes end de danske. Det vil vi prøve at rette op på. Målet med denne opgave er altså at lave et program, der kan spille Dam. For at få så mange med som muligt, vil jeg holde mig til standard BASIC 2, og arbejde i en skærmmode, som alle maskiner har mulighed for at arbejde med. Selve designet og programmeringen vil jeg dog først gå igang med i næste artikel.

Næste gang

Nu har I altså en idé om, hvad der venter jer næste gang. Hvis der er nogen, der har gode idéer, f.eks. til hvordan bruger-interface'et skal være, eller hvordan vi skal planlægge en god spilstrategi, så skriv ind så hurtigt som muligt. Det er meningen, at dette skal være et fælles projekt.

Slut for nu - over and out.

EDB på arbejdspladsen

KayJay sammenligner Mac med Arc

Her er den første artikel i serien om EDB på arbejdspladsen. Den er skrevet af ophavsmanden til artikelserien, der selv har haft en Archimedes, siden juli 1988, da den lige var kommet frem. Hvis andre vil komme med et bidrag, kan I finde annoncen på side 25 i Egebladet nummer 14.

Apple Macintosh vs. Acorn Archimedes

Da jeg for ca. 4 måneder siden blev ansat i et oversættelsesbureau, købte firmaet samtidig en Macintosh LC. Maskinen skulle, foruden tekstbehandling og andre almindelige kontoropgaver, bruges til DTP. Jeg havde foreslået, at vi skulle anskaffe en Archimedes i stedet, da jeg vidste, at den ville kunne dække de behov vi har i firmaet, samtidig med, at der er mulighed for, at levere færdige tekster i ASCII eller Word Perfect format på MS-DOS disketter, til de kunder, der gerne vil have teksten på disk. Valget faldt imidlertid på Mac'en, da ingen nogensinde havde hørt om Archimedes (Var det ikke ham grækeren med badekarret?!)


Priser

Prisen for en Mac med 2 Mb RAM og 40 Mb harddisk er lidt mindre end for en Archimedes med tilsvarende udstyr: Apple Macintosh LC 13.900/ A420 17.997, men da vi skulle se på programmer, fik jeg et chok! Horrible priser! De eneste programmer, vi anskaffede var Aldus Pagemaker 4.0 (10.100,-), MacWrite (3.350,-) og MacDraw (4.990,-). Til sammenligning kan jeg nævne tre tilsvarende programmer til Archimedes: Acorn DTP (1.562,-), 1 Word Plus (729,-) og !Draw (gratis sammen med !Edit og mange andre programmer), så alt ialt koster systemet omkring 32.000,- mod omkring 23.000,- uden printer og monitor.

Desktoppen

Macintosh har en brugerflade, noget i retning af Archimedes' desktop. Den store forskel er, at i stedet for Icon-baren i bunden af skærmen, har man en menu-linie øverst på skærmen. Selv om nogle modeller (f.eks. LC) er farvecomputere, er desktoppen ikke så farverig som på Archimedes. Hvis du har Alice in Wonderland til din Archimedes, kan du se hvordan Mac's desktop ser ud. Baggrunden består af et mønster af små mørke og lyse prikker i stedet for en ensfarvet gråtone, og vinduerne er sort/hvide. På farvemodellerne har man dog mulighed for, at give én ud af 8 farver til de icons, der bruges til programmer og filer, men kun til de streger de er tegnet med, så de er stadig fyldt ud med hvid (med mindre man vælger dem med musen, hvor farverne bliver inverteret (de hvide områder bliver farvede og omvendt)). Fordelen ved dette er, at man har mulighed for, at få fileren til at sortere dokumenter (og programmer) efter farve i stedet for alfabetisk, efter tidspunkt, efter type eller efter størrelse. De eneste icons, der er i selve desktoppen er en affaldsspand og en icon for disketter og harddisk. Man kan trække dokumenter og programmer ud i desktoppen, så man ikke behøver at åbne for directories. Både papirkurv og 'sticky' desktop findes som public domain til Archimedes. Én ting, som man ikke har mulighed for på Archimedes, er muligheden for at flytte rundt på dokumenter og programmer i vinduerne, så de står hvor som helst, man vil have dem. Dette kan føre til nogle meget uoverskuelige dokumentoversigter, hvis man ikke sørger for at have et vist system i opstillingen, men kan også give en hurtigere måde at finde de dokumenter, man er i gang med for tiden.

Musen

Musen har kun én knap, hvilket kan virke frustrerende, når man er vant til tre. Alle 

menuer ligger på menu-linien foroven på skærmen. Der er kun menuerne til ét program ad gangen, så hvis man multitasker flere programmer, skal man bruge et program, der hedder Multifinder, der gør, at man kan toggle mellem de forskellige applikationer og fileren. Uden Multifinder, kan man ikke bruge fileren, når man har startet et program op. Ulempen er, at man skal op og klikke sig frem til det program, man skal bruge, ind og finde den rigtige menu, og derefter ned i programvinduet, tit med musen holdt krampagtigt i bund. Alle menufunktioner aktiveres ved at placere musen i menulinien, holde knappen i bund, køre ned i menuen, evt. ind i en submenu, der dukker op ved siden af den forrige, finde det rigtige sted og derefter give slip fra knappen med et lettelsens suk.

En anden effekt af denne knapsingularitet er, at man som på Archimedes kan aktivere forskellige funktioner med et klik eller dobbeltklik, men der findes (f.eks. i PageMaker) funktioner, der kræver et trippelklik! Man lærer til gengæld hurtigt at bruge de genveje, der findes i de fleste programmer (de findes også i de fleste programmer til Archimedes, undtagen Draw, men nu er der kommet et Draw+, der har disse funktioner, samt mange andre af dem, der er i MacDraw inkl. libraryfunktion). Til genveje har Mac en såkaldt 'æbletast', der fungerer som en blanding af en ctrl og en funktionstast. Hvis man vil have fed skrift trykker man æble+B, kursiv=æble+I og sæt ind=æble+P. Her skal det lige nævnes, at selv om programmene er oversat til dansk, er funktionerne stadig engelske, hvilket kan være forvirrende. Hvis man er vant til engelske programmer, er det ikke noget problem.

Dette er Times 10 Punkt

Dette er Times 12 Punkt

Dette er Times 18 Punkt

Dette er Helvetica 10 Punkt

Dette er Helvetica 12 Punkt

Dette er Helvetica 18 Punkt

Dette er Courier 10 Punkt

Dette er Courier 12 Punkt

Dette er Courier 18

Programmerne

MacDraw

MacDraw ligner Draw til forveksling. MacDraw har dog nogle funktioner, som ikke er med i Draw (her vil jeg lige gentage, at der lige er kommet både Draw+ og Edit+, som har mange nye funktioner, som jeg endnu ikke har nået at afprøve). En af disse funktioner, er muligheden for at udfylde en form med et foruddefineret eller hjemmelavet mønster, der kan bruges til at skabe mange effekter. Selve opbygningen af objekter med streger er stort set den samme, men Mac har ikke de såkaldte Bezier kurver som Arc. Tekst er nemmere at arbejde med på Mac, hvor man kan rotere og ændre tekst, der allerede er skrevet, men til gengæld bliver resultatet heller ikke så pænt, men mere om fonts senere. Jeg vil dog lige nævne FontFX til Archimedes, der løser problemet med rotering og meget mere.

Rotér og flyt funktionerne er lidt mere *brugeruvenlige* på Mac. For at flytte et objekt, skal man trække i et fyldt område, hvor man på Arc kan nøjes med at trække i et hvilket som helst område inden for den boks som omgiver objektet. På Arc kunne dette give problemer ved meget tynde linier eller punkter, men det er løst i Draw+. Mac'en insisterer også på at rette op på de linier man laver, hvilket er irriterende, hvis man nu vil lave noget der er uregelmæssigt, men ellers er det meget smart. Hvis man skal vælge flere objekter på én gang, bruger man på Arc'en sin Adjust knap men på Mac skal man bruge shift på keyboardet.

På Mac kan man ikke bruge farver, hvilket ikke er

Dette er Trinity 10 Punkt

Dette er Trinity 12 Punkt

Dette er Trinity 18 Punkt

Dette er Homerton 10 Punkt

Dette er Homerton 12 Punkt

Dette er Homerton 18 Punkt

Dette er Corpus 10 Punkt

Dette er Corpus 12 Punkt

Dette er Corpus 18 Punkt

Eks. 1

noget problem hvis man ikke har en farveprinter. Arc laver de farvede områder til gråtoner på printeren, men den effekt, og mange flere, kan man få ved at bruge de forskellige mønstre, så der er scorer MacDraw et point over ArcDraw.

DTP

Macintosh har efter sigende sat standarden for DTP-programmer til andre computere. De DTP-programmer vi har til Archimedes har da også mange af de samme funktioner, hvilket man kun kan forvente af programmer der alle har samme formål. Forskellen ligger i, hvordan disse funktioner er implementeret. Jeg vil lige nævne, at jeg endnu ikke har vænnet mig helt til PageMaker, men i samme åndedrag skal det nok tilføjes, at jeg heller ikke har sat mig helt ind i de funktioner der er i Acom DTP. Dette giver mig mulighed for at sammenligne programmene på et intuitivt plan, d.v.s. af en der har lært sig selv at bruge programmene ved at prøve sig frem, frem for at bruge en masse tid på at læse manualer og den slags pjat.

Times 14 kursiv
Times 14 bold

Trinity 14 kursiv
Trinity 14 bold

Eks. 2

Som nævnt, er alle basisfunktionerne stort set de samme. Den væsentlige forskel ligger i den måde resultatet vises på skærmen (og på udskriften for den sags skyld). På Arc er det så godt som WYSIWYG, da det udskrevne bliver identisk med det man ser på skærmen (ikke 100% identisk, da der er forskel på opløsningen på skærm og printer). Mac ville jeg nærmere kalde en form for WYSIAWYGBNQ(SC%\$&£!) eller What You See Is Almost What You Get, But Not Quite (Stupid Computer grrrngh argh!). Udskriftseksemplene vist i Eks. 1, illustrerer hvad jeg mener. De er udskrevet fra PageMaker på en Apple Imagewriter II (kr. 3.900,-) hhv. Acom DTP på en Star LC10 (kr. 1.395,-). I begge tilfælde er teksten sat op, så den på skærmen står korrekt. Udskriften viser de tre mest anvendte skrifttyper i tre forskellige punktstørrelser og de tilsvarende Archimedes fonts (på Archimedes bruges der andre navne til de samme fonts). Jeg har valgt netop de tre punktstørrelser fordi Mac kun kan bruge visse

prædefinerede størrelser, uden at behøve at skalere en mindre størrelse op, resulterende i et mere kantet udseende. Læg mærke til, hvordan afstanden mellem ord og bogstaver varierer kraftigt på Mac, også selv om det så anderledes ud på skærmen.

Eks. 2 viser Times (Trinity) 14 punkt i kursiv og bold (fed), for at vise forskellen på den måde de laves. På Archimedes er kursiv og bold defineret som en separat del af en font (fylder mere på disken og kræver at de er defineret i forvejen, men resultatet bliver bedre) mens Mac simpelt hen 'vælder' skriften for at lave kursiv og fylder flere punkter på i bredden for at lave bold. Resultatet er ikke nær så læsbart.

Eks. 3 viser noget, som har givet mange problemer, når vi skulle stille regnskaber og lignende op, hvor der både er tekst, tal med og uden fortegn eller parenteser stillet op i kolonner, der ikke altid skal stå lige under hinanden. Normalt kan man stille kolonner op ved at bruge tabulator med højre, venstre eller centreret margin, men hvis kolonnerne og marginen skifter tit, kan man bruge evigheder på at stille det op. Derfor er det tit nemmere at sætte mellemrum ind, for at få rettet kolonnerne til. Som det kan ses i eksemplet, bliver resultatet så nogenlunde på Acom DTP (selv om det kan drille lidt til tider, men så har man fixed space eller 'keming' (knibning på dansk) til hjælp, hvilket jeg dog ikke har brugt i dette eksempel, da de ikke findes i PageMaker). På Mac, stod linieme lige under hinanden på skærmen, men udskriften ser lidt anderledes ud! På Arc kan man se om de skal justeres inden man skriver dem ud, men på Mac må man skrive ud og derefter gå ind og rette til på skærmen op til flere gange.

Sidste eksempel, Eks. 4, illustrerer et sidste problem. Hvis man skal bruge en hurtig udskrift, f.eks. for at lave de justeringer jeg nævnte ovenfor, kan man udskrive teksten med en lavere opløsning. På Mac kan man vælge mellem høj, standard og korrektur opløsning, men resultatet bliver meget forskelligt som man kan se. Hvor Archimedes laver færre punkter, men beholder størrelsen på bogstaverne, laver Mac'en simpelt hen skriften mindre. Den prøver at holde linie-længden den samme, så derfor bliver mellemrummene mellem ordene meget varierende. Imagewriter er noget hurtigere end en LC10, så derfor vil

man sjældent bruge denne mulighed, men da selv tekstbehandlingsprogrammerne bruger computerens fonts i stedet for printerens indbyggede fonts, kan det tage lang tid at skrive sin tekst ud.

PageMaker har, til forskel fra Acorn DTP, indbygget tekstbehandler, med mulighed for at definere tvunget side-, spalte- og afsnitsskift. Det har nogle af de andre DTP programmer til Archimedes også, men jeg har ikke selv prøvet at arbejde med dem. Dog har jeg fundet ud af (Tak Olav. Åh det var såmænd så lidt, red.), at man kan lægge kommandoer til Acorn DTP direkte ind i sin tekstfil, inden man importerer den.

1. linie Her står en tekst
2. linie Dette står lige under
sidste linie Er de lige?

Fonts

Mac anti-aliaser ikke sine fonts, heller ikke i udskriften, med det resultat, at bogstaverne bliver meget kantede. I PageMaker bliver alle tegn, der er under en vis størrelse komplet ulæselige, og hvis de er meget små (ca. 12 punkt), bliver de slet ikke tegnet, men i stedet bliver der lavet en 'shaded' bjælke, hvor teksten skulle stå. Derfor bliver man nødt til at forstørre siden op, så man kun kan se en del af teksten af gangen. Dette

Denne linie er skrevet ud med høj

Denne linie er skrevet ud med standard

Denne linie er skrevet ud med korrektur

hjælper ikke på overskueligheden. Hvis man bruger en laserprinter, bliver udskriften dog pæn, da man bruger printerens indbyggede fonts.

Der er en ting ved Mac som jeg er meget misundelig over, nemlig muligheden for at koble en A4-størrelse høj-opløsningsmonitor til. Der kan man se en hel side på én gang, hvilket er perfekt til DTP. Den koster godt nok et halvt parcelhus, men er u-undværlig (sagde sælgeren), hvis man skal lave seriøs DTP (jeg påpegede, at hvis man virkelig var seriøs, anskaffede man sig en Archimedes ("En hva' for en !?! Er det en PC eller så'n?"). Suk!).

1. linie Her står en tekst
2. linie Dette står lige under
sidste linie Er de lige?

Eks. 3

Den vigtigste ting ved Mac'en, set fra en købers synspunkt, er, at den virker meget overskuelig, og man kan (igen ifølge sælgeren) lære at bruge Mac'en fuldt ud på 33 sekunder med bind for øjnene og med højre fod i en spand vand, eller noget i den retning. For en programmør/hacker/entusiast eller hvad nu en gennemsnits Acorn ejer kalder sig, virker den lidt 'lukket', d.v.s. at der ikke er mulighed for at forlade desktoppen og lege med supervisoren eller BASIC'en eller rode med *modes* og *taskmanager* eller bare fikle lidt

Denne linie er skrevet ud med høj

Denne linie er skrevet ud med medium

Denne linie er skrevet ud med lav

Eks. 4

rundt i programmerne og prøve at modificere dem osv. Man kan selvfølgelig købe forskellige programmørværktøjer o.l.

For en *teknofob* tror jeg at Archimedes er lidt mere skræmmende. Der er mulighed for, at komme til at ændre på konfigurationen eller skifte til et skærmmode, så man ikke kan komme tilbage til den gamle setting uden at skulle resette maskinen helt, men den slags ting gør livet lidt sjovere for os computerfreaks, eller hvad?

Alt ialt, hvis jeg skulle vælge mellem de to systemer til de behov vi har i vores firma, ville prisen på hardwaren ikke blive afgørende, men prisen på softwaren ville nok få noget at sige. Set fra et rent teknisk synspunkt ville valget helt afgjort falde på Archimedes, selv om den fylder mere i landskabet. Desværre er der en ting, der gør Macintosh mere attraktiv, nemlig at de fleste trykkerier, reklamebureauer etc. bruger Macintosh (eller PC). Man kan spare mange penge til opsætning, ved at sende en diskette med en færdig DTP-fil til trykkeriet, hvilket mange af vore kunder er glade for. Man vælger selvfølgelig et trykkeri med samme anlæg som man selv bruger, hvilket er nemt hvis man bruger Macintosh eller PC. Hvis man har valgt Archimedes, har man et lille problem: så vidt jeg ved, er der ikke i Danmark et eneste trykkeri, der bruger Archimedes, men hvis der er, vil jeg meget gerne vide det. Hvis man på Archimedes havde mulighed for at gemme et dokument i PageMaker format ville problemet være løst, men indtil videre er der åbenbart ingen der har tænkt på det. Vi kan jo allerede køre IBM programmer (nu også med EGA/EGA+ grafik) og endda også Spectrumprogrammer.

Konklusion

Min konklusion må være, at Apple Macintosh er et udmærket værktøj, som man kan lave professionelle tryksager på, uden meget kendskab til bogtrykning, men hvis man kan undvære danske programmer, og har mulighed for at få udskrevet reproklart materiale uden bogtrykkerens hjælp, så vil Archimedes i det lange løb være både den billigste og bedste løsning.

Ann's alder

Der var flere der havde sendt løsningen til den opgave, der var med i nummer 14. De tre heldige vindere var 6551 Jens Bertelsen, 6519 Jan Magnussen og 6612 Bjørn Petersen.

Jens havde indsendt en længere matematisk udredning af problemet inklusiv en grafisk afbildning af løsningen, Jan havde lavet en løsning i form af to ligninger med to ubekendte (stort set det samme som Jens), og Bjørn havde lavet et BASIC program, der løste problemet.

Jeg vil vise den måde som Jan brugte, som iøvrigt er den metode, jeg selv har brugt:

$x = \text{Ann's alder}$

$y = \text{det antal år, der er gået siden Mary var lige så gammel som Ann er nu.}$

Derved får vi to ligninger:

$$24 = 2(x - y) \text{ og } 24 - y = x$$

Ved at forkorte og addere de to ligninger får vi:

$$x - y = 12 \text{ og } x + y = 24 \text{ eller } 2x + 0y = 36,$$

altså $x = 18$, Ann's alder er 18 år.

De tre vindere har fået tilsendt nogle programmer til deres computere.

Der var ingen der vidste, at gåden stod i Piers Anthony's "Shade of the Tree". Han skriver ellers nogle af de bedste *science fiction* og *fantasy* bøger, der findes, selv om ovennævnte var hans første (og eneste) horror bog til dato. Læs dem!

Der var forresten en anden gåde indbygget i artiklen. Hvis du synes, at Duva Thunder var et dumt navnt, så læg mærke til navnet på hans tante Anna Gram. Et anagram er som bekendt en sætning eller et ord, hvor bogstaverne er byttet rundt for at danne et nyt ord eller sætning (eller, som her, et navn). Hvis man bytter rundt på bogstaverne vil man kunne danne sætningen, "du har vundet", og ville, hvis man havde sendt ind, have vundet et eller andet meget dyrt og dejligt.

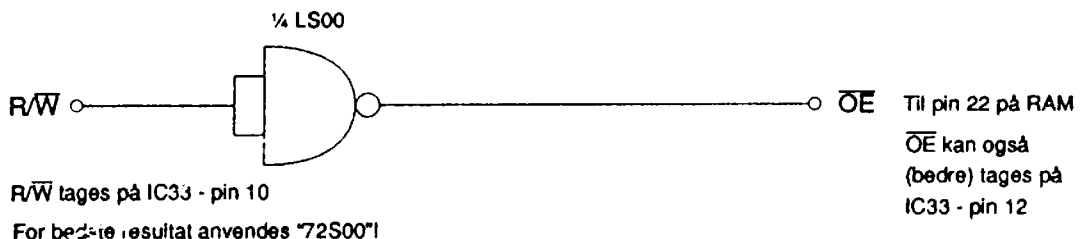
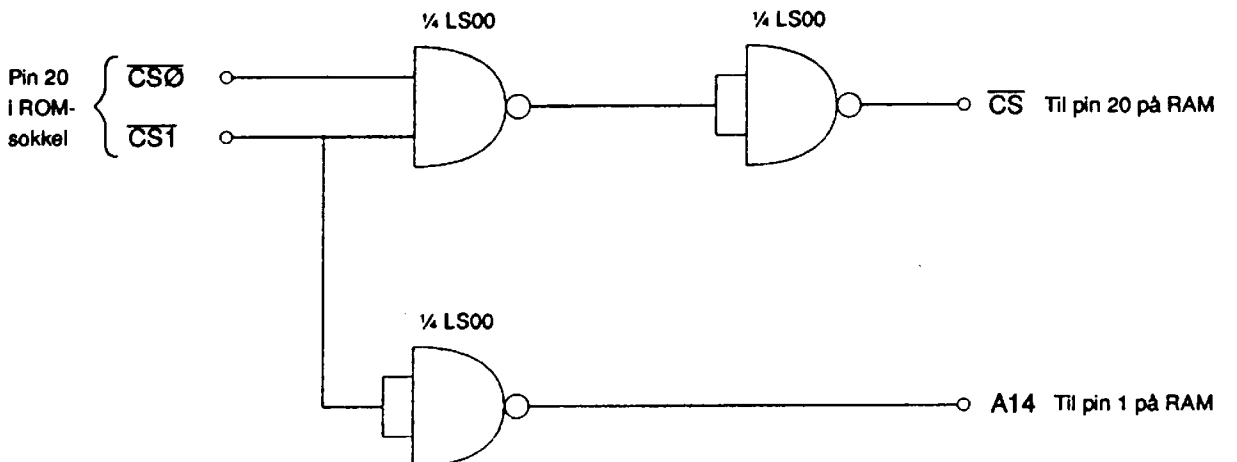
MORALE: læs altid bladet godt igennem og skriv altid ind, hvis du ved noget, som andre ikke ved!

Sideways RAM

Lars fortæller om sidelæns RAM på B'ere og Master'e

Som bekendt har den gode gamle BBC-C (hvaforenhest? red.) ikke for meget RAM at rutte med, og dette afstedkommer da også en del problemer for f.eks. spilprogrammører men i særdeleshed også for folk som gerne vil have residente utilities i deres maskine. For BBC-B har den officielle løsning fra Acorns side været, at man lagde sine residente programmer i EPROM, hvilket er en udmærket løsning. Men da man designede BBC+128 og derefter BBC Master tilføjede man en utrolig praktisk detalje: Sideways RAM.

Hermed blev det muligt at have sin ROM-baserede software til at ligge på diskette/disk og man kunne let selv lave residente programmer, som ikke kolliderede med andre programmer. Det er faktisk muligt at udvide sin BBC-B med sideways RAM, uden at det skal koste en formue, som det ellers vil gøre, hvis man køber 16KRAM i England. Der er imidlertid et par ting man skal tage højde for: (Bliv ikke bange for den følgende tekniske forklaring - man kan godt springe over...)



På nær de viste pin-forbindelser, er RAM-kredsens pin-layout som en 16K-EPROM (ROM)

Da BBC-B kun er forberedt på ROM er det ikke muligt blot at erstatte en ROM med en tilsvarende RAM-kreds, da bl.a. R/not W mangler i ROM-soklerne. Dette er dog ikke et problem, da den kan tages andre steder i maskinen, helst så logisk tæt på CPUen som muligt, da for mange gates mellem CPU og RAM introducerer et delay. Et godt sted at tage R/not W er på IC33 pin 10, selvom der allerede her er introduceret et delay på 19 nS (2*9,5nS for 741s04).

Problemer kan også opstå, hvis du forsøger at anvende sideways RAM sammen med et ROM-board. Det første problem kan bestå i, at databussen på ROM-boardet kun er bufferet fra ROM mod CPU. Hvis dette er tilfældet kan man selvsagt ikke skrive til SWRAM på ROM-boardet.

Et andet problem kan opstå, hvis ROM-boardet er forbundet til BBC-grundprintet via link S21. Før S21 er der 2 gates som udreder ROM-select. Mange ROM-boards indkobler endnu en gate i S21 før ROM-select signalet tilbage til grundprintet. Dette kan i mange tilfælde betyde et delay, som er stort nok til at SWRAM ikke kan fungere, hverken på ROM-board eller på grundprintet.

Med andre ord: Det er let at sætte SWRAM direkte i soklerne på grundprintet, så længe der ikke er et ROM-board i maskinen! De fleste SWRAMer man kan købe er 16K versioner med eller uden batteri-backup og writeprotect. Prisen er typisk omkring £30 (ca kr.420,-), hvilket er USANDSYNLIGT dyrt for så lidt. Det er muligt at lave sin egen 32K SWRAM, d.v.s. den vil optage 2 sokler, fysisk og logisk, for kr.100,- til kr.200,- afhængig af dagsprisen på RAM. Man kan således have BASIC til at sidde i soklen længst til højre, DFS/ADFS i soklen længst til venstre og 32K SWRAM i de to midterste sokler.

Når jeg skal lave en 32KRAM anvender jeg en kreds som hedder 43256C-12L. Dette er en CMOS RAM med 256Kbit (32Kbyte) med en accesstid på kun 120nS, hvilket snildt rækker til Beeb'en. Men pinlayoutet er ikke det samme som for en ROM-kreds: Bl.a. er der kommet en ekstra adresselinie samt R/not W.

For at lave chip select signale (not CS) fra de to ROM-sokler om til en ekstra adresselinie anvender jeg 3 NAND-gates, helst en 74s00 som har et delay på kun 3nS mod 741s00'ens 9,5nS. Se

diagrammet for mere information.

Output Enable (not OE) på RAM-kredsen kan tages i ROM-soklens pin 22 eller via den resterende NAND-gate, som anvendes som inverter på R/not W.

Jeg er igang med at designe et ROM/RAM-board med 8 sokler til ROMer og 4 sokler til hver 32KRAM. Dette board med 128K statisk RAM kommer således til at erstatte soklerne på grundprintet fuldstændigt.

For tiden har jeg en noget speciel opbygget computer med et gammelt 8 sokkel ROM-board, en del ROMer og 112K sideways RAM. Det virker, men det ser underligt ud... (jeg tror jeg ved, hvordan man opfandt wire-wrap!).

Imidlertid har jeg nu fået brug for en del utilities til at håndtere al den RAM. En nyttig rutine (*PLUG) er en installerings rutine, som automatisk loader en ROM fra diskette ind i SWRAM, enten på første frie plads, eller på en forud valgt plads.

En anden nyttig rutine (*MAPMEM) returnerer en liste over den tilgængelige SWRAM i maskinen, enten al RAM eller kun det som ikke er optaget af en ROM.

Begge rutiner er på klubdiskette nr. 3.

Når man har fået installeret sin SWRAM, kan man begynde at lege med sine egne residente programmer og ROM-filing systemet.

Når programmer ligger i ROM-området er der ikke så stor fare for, at andre programmer smadrer dem (død over UIM) og dermed minimeres faren for crashes.

For eksempel kan man om-arrangere sit keyboard på denne måde, så [Ø] kommer til at sidde de rigtige steder, uden at der er konflikt med andre programmer. Evt. kan du bytte om på funktions-tasterne i VIEWWORD (VIEWWEIRD!) så de sidder mere logisk!

På klubdiskette nr. 3 har jeg lagt en ROM (R.MAP) som kan installeres med følgende kommando:

*PLUG R.MAP <ENTER>



Tastaturet er nu noget á la Archimedes' layout. Desuden er der mulighed for at lave nye tastekombinationer med SHIFT+CTRL, som enten kan generere et tegn eller kalde et program gennem en vektor. Således er det muligt at lave en resident lommeregner, note pad eller hvad som helst, som kan kaldes frem med en HOTKEY kombination. Prøv forresten at trykke 'SHIFT' + 'CTRL' + '1'(virker ikke i MODE 7).

For de som ikke har SWRAM ligger der også en version af dette program, som kan køre i base-memory (\$.MAP). Sourcecoden ligger der også (\$.KEYMAPS). Denne kan både generere \$.MAP og R.MAP.

Endvidere ligger der på disketten en ROM (R.VOICE), som indeholder noget samlet lyd, som afspilles ved forskellige hændelser, f.eks. en opstået fejl.

Hvis du opgraderer din Beeb med SWRAM forøger du din computers styrke mængdefold, specielt hvis du har en disktestation/harddisk. Og til prisen er det vel ikke at kimse ad! Tænk, nu kan du køre UIM og andet software, som sætter sig ud over Beeb'ens værste begrænsning -hukommelsen!

*PLUG og *MAPMEM virker også på BBC Master.

Cambridge-Pascal

- en vindues-baseret pascal-oversætter.

Jeg er blevet betroet at anmelde en pascal-oversætter med navnet "Cambridge Pascal Compiler" fra firmaet Solent Computer Products Ltd. Oversætteren, der ifølge Royal Data koster ca. kr. 1000.-, ankom i en slags plastic-box, der i form minder om et video-bånd.

Boxen indeholder

1 stk 3.5" diskette med oversætter og eksempler

Manual i ringbind, 60 sider i ca. A5 format

Bruger-registreringskort

Det første indtryk er gunstigt, installationsproceduren er simpel. Man kopierer simpelt hen hele disketten ind på sin harddisk, hvis man har sådan en, ellers laves en kopi af den originale diskette og oversætteren køres fra denne diskette.

Disketten indeholder tre kataloger:

- !Pascal ;oversætter-application
- !PasTrace ;facilitet til at trace et program
- Examples ;mange, udmærkede eksempler

Oversætteren er beregnet til at køre i wimp-miljøet, men kan også køres fra kommando-

promptet. Oversætteren startes ved at klikke to gange på !Pas-iconen; kort efter har oversætteren installeret sig på programmenuen nederst på skærmen sammen med Taskmanager osv. Ved at klikke menu på denne icon kan du vælge mellem quit, info, options_menu og at gemme nuværende options. Man kan nu oversætte et Pascal-program ved at trække fil-iconen ned på oversætterens icon.

Options er forskellige tilstande, man kan slå til og fra:

listing ;programudskrift under oversættelsen


auto run ;start programmet, så snart det er oversat

trace ;generer kode, der muliggør senere at singlesteppe programmet

ram ;oversæt hele programmet i hukommelsen

extensions ;inkluder udvidelser fra ISO-standarden, f.eks grafik

quick ;generer kode for maximal udførselshastighed

Desuden kan man sætte en parameter "Code Limit", dvs. en grænse for hvor stort 

programmet kan være. Oversætteren består af flere dele, her angivet med den hukommelsesplads, de ca. kræver under kørslen:

Compiler Module	162 kbytes
ABCLibrary Module	19 kbytes
CPCLibrary Module	17 kbytes
FPEmulator Module	34 kbytes
!Runimage Application	64 kbytes

	296 kbytes

Oversætteren er skrevet i BBC Basic og oversat med ABC-Compileren, derfor er de to libraries nødvendige. Det er selvfølgelig kun nødvendigt at bruge FPE, hvis programmet opererer på reelle tal. Selve oversættelsen kan der siges både godt og ondt om. Oversættelsestiden er absolut ikke imponerende - langsom er et mere præcist udtryk. Hvad der gør dette værre endnu er, at der ingen mulighed er for at lave objektkode, dvs. oversatte programstumper, som kan inkluderes i programmer, uden at skulle oversættes igen. Dette er noget af det første, der bør ændres i en ny version.

Til gengæld kører de programmer, oversætteren genererer, meget hurtigt. Det er mit indtryk, at koden er lynhurtig såvel som meget kompakt. Mange eksempler fylder ikke mere end et par kilobytes i oversat form - det kan man absolut ikke opnå med Acorns Pascal!

Pascals historie er tæt knyttet til undervisning. I sin tid, da Niklaus Wirth opfandt sproget, var det meningen, at klarhed skulle være den ledende tråd i sprogets syntaks. En masse avancerede faciliteter ville ikke medføre et mere pædagogisk sprog, og derfor har Pascal-standarden i dag kun ganske få og rudimentære i/o-faciliteter. Derfor bruger man sjældent standard-Pascal til egentlig systemudvikling. På den anden side er der meget få programmører, der ikke kender sproget. Derfor er det let at udveksle programskitser - pseudokode er for mange noget, der i mistænkelig grad minder om Pascal.

Cambridge Pascal indeholder heldigvis en mængde udvidelser til den almindelige ISO-Standard. Det drejer sig i korte træk om grafik, streng- og filbehandling samt kald til operativsystemet. Alle disse ting er det muligt at lave i sine programmer uden at ty til maskinkode, som man skal med Acorn Pascal. At indlejre maskinkode i

sine programmer er til gengæld ikke muligt; man kan dog slippe uden om det ved at bruge swi-kald fra sit program (hvis man nu absolut skal...). Oversætteren kan desuden lave moduler, hvis man indsætter passende instrukser i starten af sit program.

Manualen gør ikke noget forsøg på at undervise i Pascal, den er kun til at slå op i. Teksten er kort og klar og omhandler hovedsageligt udvidelser af sproget. Man kunne dog ønske sig et indeks bagest, selvom der kun er ca. 60 sider at fare vild i.

Eksemplerne på originaldisketten dækker mange af de forskellige muligheder, oversætteren giver. De drejer sig om grafik, moduler, benchmarktests plus en masse mere eller mindre klassiske godbider. De har alle præg af at være demonstrationsprogrammer, små og instruktive.

Ud over de nævnte, er der også et eksempel på brugen af interface til Wimp-miljøet. Dette er et procedurebibliotek, som kan inkluderes i dit program. Næsten alle wimp-kald er dækket, de sidste kunne man evt. lave selv. Måske kan Pascal være et bedre sprog til dette formål. Wimp'en er et lidt rodet miljø - klarhed i koden er derfor efterstræbelsesværdig.

Er du Pascal-fan, må jeg så absolut råde dig til at overveje købet af Cambridge Pascal. Du får på denne måde mulighed for at udnytte alle din maskines muligheder uden, at du er nødt til at lave alle mulige krumspring for at gøre dette. Dertil kommer, at det er legende let at anvende oversætteren, man kan derfor koncentrere sig udelukkende om sit program. Hvis nogle edb-skoler skulle have Archimedes-maskiner, bør de overveje at anskaffe denne oversætter.

De minusser, jeg retfærdigvis må nævne, er oversættelsestidshastigheden, den manglende brug af objektkode + en eller anden form for linkning.

Konklusionen må være, at der er tale om et absolut godt produkt. Hvis man for de 1000,- kr også fik "turbo"-speed og evt. mulighed for at lave objektkode, ville det måske være et vinderprodukt...

Hans Otto Lunde

Adventuresiden

Martin Søndergaard fortsætter med løsningen til Enthar Seven

Hej igen. Her kommer forklaring til den sidste del af eventyret "Enthar Seven". Efter turen til registreringen synes jeg du skal indtage område 3.

Du skal bruge følgende ting: et bolsje, en hjelm, en iltflaske og en lampe. Husk at forbinde iltflasken med hjelmen, da visse dele af området er uden ilt. Du befinder dig midt i en skov, som jeg vil lade dig om at kortlægge. Her skulle du kunne finde noget gemt under bladene. Det er en underlig platform eller plade, der anbragt i skovbunden. Det ser ud som om den er lavet til et par fødder, så prøv at stille dig på den. Her er området uden ilt, du kan kun gå et par skridt uden hjelm, men så må du også tage den på.

Det stillads du går på, har vist set bedre dage, det er jo faldet sammen 3 steder. Et af stederne kan du springe over, et andet sted kan du svinge dig over ved hjælp af en slyngplante, og så kan du bruge en planke på vejen tilbage, det tredje sted må du også bruge planken. For at åbne døren, skal du undersøge tragtens diameter og diameteren på bolsjet, de skal passe sammen. Når du er indenfor døren, kan du godt tage hjelmen af, du skal jo også bruge ilt på tilbagereturen. Nu får du et problem, du bliver nemlig fanget af en edderkop, der åbenbart har forgiftet dig, for du har et frygteligt mareridt, så du må hellere vågne op, inden du dør af skræk, men det kun for at konstatere, at du er bundet med edderkoppespind på hænder og fødder, og alle dine ting er borte. Hvis du undersøger væggen, vil du opdage spidsen af et søm. Prøv at gnide spindet på sømmet, du skal prøve op til flere gange, men endelig kommer du fri. Du kan kun gå sydpå, bare for at opdage at alt er totalt mørkt, men så kan jeg fortælle dig at selv om du ikke kan se dem, så ligger alle dine ting her, du skal bare tage dem.

For at komme ud, skal du igennem en mangefarvet labyrint; hertil vil jeg kun sige, at bolsjet egentlig er en forvandlingskugle. Husk stadig at

tage alt du finder. Når du igen står i hovedbygningen, skal du helst have følgende: en vifte, en Y-formet gren, en planke, en hjelm, en lampe, resterne af et bolsje og en medicinflaske med gift. Så er du parat til område 5.

Her skal du bruge en åre, en medicinflaske, en bidetang og den Y-formede gren. Man kan vist godt sige, at dette er helvede på jorden. Du befinder dig midt ude i en meget ugæstfri ørken. Det er heldigt at du har den Y-formede gren, og til sidst ender du da også ved bredden af en flod, men her må du hurtigt skjule dig. Jeg vil foreslå, at du gemmer dig under hækken mod vest. Vent her indtil de fjendtlige jægere er gået forbi, så kan du låne deres kano. Hvis du kryber lidt længere ind i hækken, vil du finde den mest værdifulde genstand på hele Enthar Seven, det er et meget besynderligt halssmykke, og det vil du sikkert give mig ret i senere. Når jægerne er forsvundet, skal du forsøge at styre kanoen ned ad floden, så godt du kan.

Efter havariet samler du alle dine ting sammen og klæder dig på, og prøv så at finde en lille hytte og skaft dig adgang til den. Derefter skal du finde den kødædende plante, som du dræber med giften, resten af ruten er bare a piece of cake.

Back to Headquarter. Inventarlisten efter denne tur: En skruetrækker, et halssmykke, en åre, en tom medicinflaske, en bidetang, en Y-formet gren og et halstørklæde. Inden du går videre til område 2, gå ned og fjern spejlet fra væggen. Du skal bruge skruetrækkeren og puden. Vær nu forsigtig.

Nu mangler kun område 2, men det består faktisk af 2 områder, men lad os nu bare tage dem et ad gangen. Tag tapen med. Du starter på fortovet udenfor en herskabsvilla eller noget i den stil. Gå op ad trappen og ring på døren. Nu forstår du måske, hvorfor du skulle igennem registreringen. Undersøg nu hele huset. Hvis du kravler op

på et af bordene vil du opdage en lem i loftet. Her ligger en kortsluttet ledning, den skal du reparere med tapen. Når du har undersøgt hele huset, gå ned til computeren i kælderen, det var strømmen til den du fiksede oppe på loftet. Der står 2 ord på skærmen, det er kommandoer til teleporten, det vil sige at du enten kan komme tilbage til hovedbygningen eller ud til Yarral-bjergene. Lad os tage tilbage til hovedbygningen. Nu skal du helst være i besiddelse af følgende: en kiks, et dokument (testamente), en manual, en farveblyant og en dagbog. Der kun en side tilbage i dagbogen, der er aftryk af skrift på siden, så prøv at gøre ligesom detektiv gør i bøgerne, brug farveblyanten på siden, det skulle bringe teksten frem. Manualen giver dig en forklaring til halssmykket.

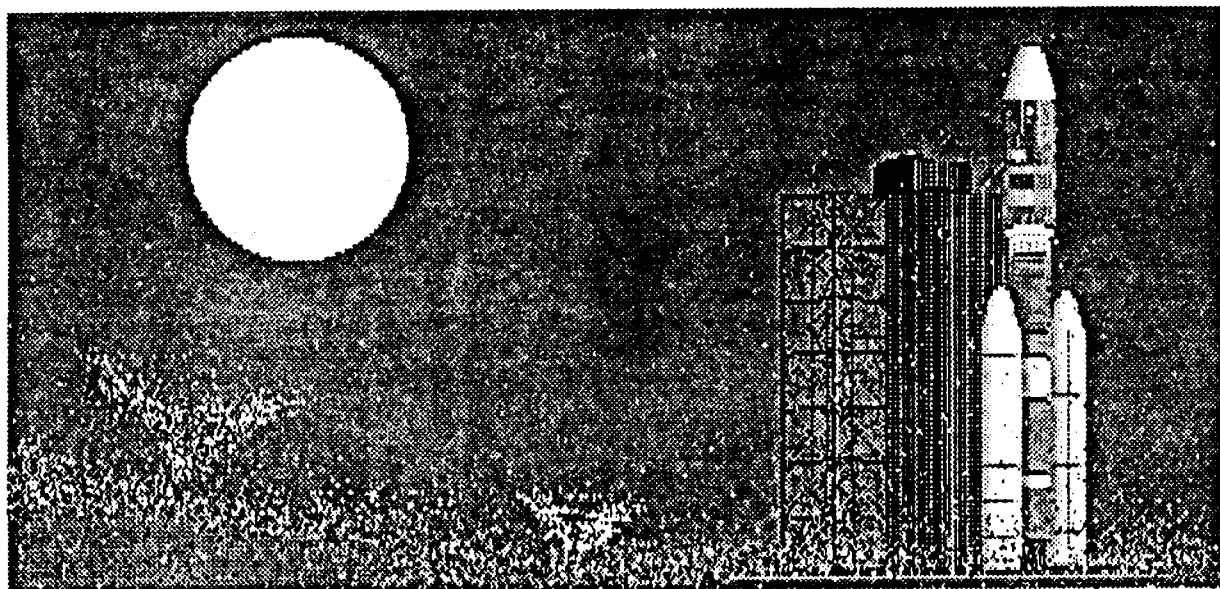
Ok, lad os smutte tilbage til teleporten i kælderen og derfra ud til bjergene. I første omgang skal du bruge lampen og hammeren, du kan også tage halssmykket på, så går det hurtigere med at komme tilbage hertil. Først skriver Yarryl på computeren, hvorefter du går ind i teleporten. Du befinder dig for foden af bjergene. Følg stien indtil du kommer til en hule, du må hellere tænde lampen, ellers kommer du bare galt afsted. Grøften er til at hoppe over, du skal bare huske, at du ikke skal bære rundt på for meget, det afkorter længden af dit spring. Muren skal smadres med hammeren, hvorefter du finder en gammel skrift-rulle, der fortæller vejen gennem bjergene. Teleport tilbage til hovedbygningen. Du kan ikke læse kortet uden oversætteren. Nu skulle du være parat

til turen gennem Yarryl-bjergene.

Du skal bruge følgende: et spejl, en kiks, halssmykket (om halsen), en flaske med vand, blomsten og resten af bolsjet. Kortet starter fra lokaliteten efter skiltet. Husk at spise undervejs. Aktiver smykket før du starter, da du ellers ikke kan komme tilbage igen.

Når du er kommet gennem bjergene, skal du først gå ind i Meta Lorvox's. Her finder du en pille, der kan beskytte dig mod de fleste sygdomme, du må hellere spise den, husk at skylle efter med vand. Nu skal du finde graven med kaptajn Benjamin Lorvox. Åben døren og gå ind, når du kommer til stedet med laserstrålen, tager du brillerne på og reflekterer strålen i spejlet, det skulle ødelægge laseren. Snig dig så forbi robotten uden at den ser dig: Du skal op i den nordlige ende af gangen, her er der nemlig en sidegang, smut ind ad den. Et stykke inde finder du et mere eller mindre opløst lig. Det har en Memomatic i hånden. Tag den og teleport tilbage til hovedbygningen. Nu kan du åbne pengeskabet, og du kan tage kortet til at åbne døren ud til rumskibet. Før du går ud skal du huske at tage halstørklæde og briller på, da vejret ikke er for godt derude. Det er det eneste du skal have på dig, ellers går det bare galt.

Slut, prut, finale.



 *Royal Data ApS* 

afholder nu:

SOMMERUDSALG

**Ring eller skriv efter
komplet tilbudsliste**

Stærkt nedsatte priser!

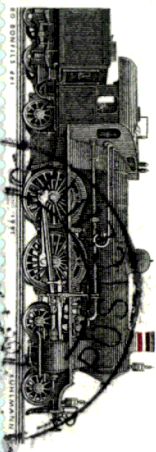
Ved bestillinger på under kr. 1000, opkræves et ekspeditionsgebyr på kr. 25.

Royal Data ApS, Sundvej 34, Hvalpsund, 9640 Farsø

Tlf. 98 63 81 50 / 98 63 81 93 oplyser om nærmeste forhandler

DANMARK LITRA P 1907

4.75



6541
Ivan Helsinghof
Syrkensvej 28
2300 København S

Aisender:

EGEBLADET
V/ Karsten Jensen
Victoriagade 12
DK-9400 Nørresundby

Videresendes ikke ved vedvarende adresse-
ændring, men returneres til "Egebladet" med
oplysning om den nye adresse.