



Redaktionelt	3
Iterationer, Mandelbrot-mængden og komplekse tal	4
CP/M-plus efterlysning	7
Brugervejledning til Mandebrot-programmet.....	8
C++, Objektorienteret programmering i C	9
PC-volumes	
PC-SIG 460 Alberta agricultural programs	11
PC-SIG 461 Ridgetown College programs	11
PC-SIG 462-463 CK System	12
PC-SIG 464 PBASE 1.2	13
PC-SIG 465 Family ties	13
PC-SIG 466-468 CPA - Ledger	14
PC-SIG 469-470 Mr. Bill	15
PC-SIG 471 Present 5.1	16
PC-SIG 472 Simplified business bookkeeping	18
PC-SIG 473-474 Trivial towers	19
PC-SIG 475 Monopoly PC / Tune Trivia	20
PC-SIG 476 Patrick's best games	21
PC-SIG 477 Name gram / Break down / Fone word	22
PC-SIG 454 UniForth	18
Uddrag fra bulletin board konference - Pascal	24

Møder

Der er endnu ikke planlagt møder for efteråret 88.

INDMELDELSE OG KONTINGENT

Indmeldelse i MUG Denmark kan foretages ved indbetaling af kontingent (pt. 200 kr. årligt) på giro 5 68 65 12, MUG Denmark, ved Jørgen Petersen. Der opkræves ikke indmeldelsesgebyr.

Oplag: 300
Tryk: Dansk Tidsskrifts Tryk.
Redaktion: Leif Olsen.

Redaktionen afsluttet 25/4-88.

Næste nummer forventes udsendt i løbet af juni-juli. Indlæg eller annoncer til bladet sendes til redaktionen (Leif Olsen). Indlæg, der er på mere end 2 sider, helst på diskette eller via Bulletin-Boardet. Indlæg honoreres med et volume fra et af klubbens biblioteker.

Redaktionelt

MEMLEMSDISKETTE(R)

Så er der igen medlemsdisketter på vej ud og denne gang hele TO styk. På den ene findes bl.a. det program til "tegning" af mandelbrot-mængden, som Jørgen Petersen demonstrerede på medlemsmødet i april.

RAM-KREDSE

Kassereren ligger pt. inde med et mindre lager af RAM-kredse (256 Kbit, 150 nsec), som sælges i klumper af 8 for 350 kr. Dette er et pænt stykke under den nuværende markedspris på ca. 65 kr/stk. plus moms!

DISKETTE-PRISER

Da kassereren efterhånden er gået helt over til kun at sælge een type disketter, er restlageret af kulørte disketter på tilbud til kun 100 kr for 10 stk. Se iøvrigt sidste side i bladet.

MODEM-KREDSE

Såfremt enkelte medlemmer ligger inde med modem-kredsen AM7911 (fra MPS-gruppens modem-konstruktion) og gerne vil afsætte den, har Erk Martinsen tilbudt at formidle kontakt til interesserede radio-amatører. Erk Martinsen kan træffes på tlf: 02 18 48 32.

Leif Olsen

Mandelbrot

Iterationer, Mandelbrot-mængden og komplekse tal.

Der var sikkert mange der så Tor Nørretranders' HVÆLV udsendelse om fraktaler i efteråret, og nød de mange fascinerende og farvestrålende billeder af Mandelbrot mængden - eller rettere fra Mandelbrot mængdens grænseområde. Der er sikkert også mange der har prøvet at generere lignende billeder selv, alligevel vil jeg skrive lidt om hvordan man udfra det meget simple udtryk:

$$z_{n+1} = z_n^2 + c \quad (1)$$

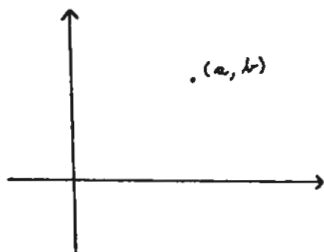
kan nå frem til disse fantastiske billeder.

I denne sammenhæng vil vi altid lade den første værdi af z (z_0) være 0 (nul) således at:

$$\begin{aligned} z_1 &= 0^2 + c \\ z_2 &= c^2 + c \\ z_3 &= (c^2 + c)^2 + c \\ &\cdot \\ &\cdot \text{ osv.} \end{aligned}$$

Enhver kan overbevise sig om, at selv for ret små værdier af c vil z_n lynhurtigt blive et meget stort tal, som de fleste computere vil have svært ved at håndtere, så man kan jo godt undre sig over hvordan man kan få sådan en billedrigdom ud af et så "dødsygt" udtryk. Det bliver da også nødvendigt at gå lidt udenfor vores hverdags erfaringer hvad angår tal og regning, nemlig til de komplekse tal.

For at kunne arbejde med de komplekse tal må vi indføre størrelsen i , hvor i har den egenskab at $i \cdot i = -1$ eller $i = \sqrt{-1}$. Et komplekst tal z er et tal af formen $a+ib$, hvor a og b er tal som vi kender dem (3, 4.562, 6.1, ... osv.) og i er den netop indførte størrelse. Man kan også tænke på det komplekse tal $a+ib$ som punktet (a,b) i et almindeligt koordinatsystem; a kaldes for realdelen og b for imaginærdelen af det komplekse tal. Reglerne for regning med komplekse tal minder meget om de regneregler vi kender, når vi bare husker at $i \cdot i = -1$. Det vi får brug for her, er at kunne addere og multiplicere komplekse tal.



Hvis vi har de to komplekse tal $a+ib$ og $c+id$ gælder:

$$(a+ib)+(c+id) = (a+c)+i(b+d)$$

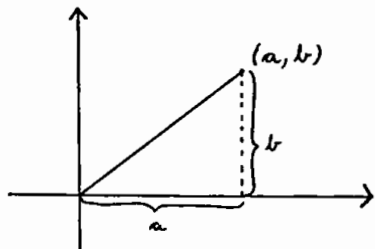
og

$$\begin{aligned}(a+ib)(c+id) &= ac+aid+ibc+ibid \\ &= (ac-bd)+i(ad+bc)\end{aligned}$$

eller som vi får brug for det her:

$$\begin{aligned}(a+i)^2 &= (a+ib)(a+ib) \\ &= aa+aib+iba+ibib \\ &= (a^2-b^2)+i(2ab)\end{aligned}$$

En sidste vigtig ting vi får brug for her, er at kunne bestemme "størrelsen" af et komplekst tal. Det er måske nemmest at tænke på størrelsen af et komplekst tal $a+ib$ som længden af det liniestykke, der forbinder punkterne $(0,0)$ og (a,b) i et koordinatsystem som omtalt ovenfor. Denne længde er jo som bekendt (pythagoras):



$$\sqrt{a^2+b^2}$$

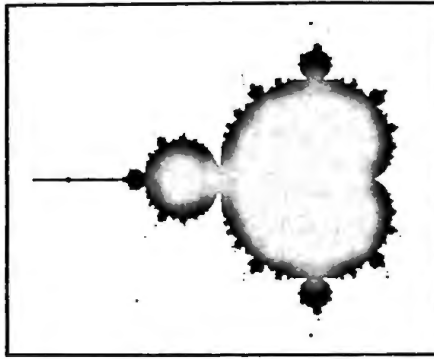
For et komplekst tal $a+ib$ skrives denne størrelse således:

$$\begin{aligned}|a+ib| &= \sqrt{(a+ib)(a-ib)} \\ &= \sqrt{a^2+b^2}\end{aligned}$$

Mandelbrot mængden (opdaget af matematikeren B.B. Mandelbrot i 1980) er mængden af alle de komplekse tal, der opfylder den betingelse, at størrelsen z_n i udtrykket (1) aldrig bliver større end 2, lige meget hvor stor n bliver. Størstedelen af de komplekse tal, der tilhører Mandelbrot mængden, ligger indenfor området:

$$\begin{aligned}-2.25 &< \text{realdelen af tallet} < 0.75 \\ -1.5 &< \text{imaginærdelen af tallet} < 1.5\end{aligned}$$

Mandelbrot mængden er den noget knortede cirkel-agtige figur, der er gengivet på figuren på næste side.



Det vi nu kan gøre hvis vi skal lave et af de meget farverige billeder som jeg omtalte i indledningen, er at vælge et område der ligger indenfor det før omtalte område (evt. vælges hele området). Områder, der indeholder noget af kanten af Mandelbrot mængden, giver de mest interessante billeder; f.eks. kunne man vælge området hvor:

```
-1.781 <= realdelen      <= -1.764
 0.0   <= imaginærdelen <= 0.013
```

Det område vi har valgt inddelles nu i et passende antal punkter, f.eks. svarende til opløsningen på den skærm vi ønsker at vise det færdige billede på. For hvert punkt skal vi nu vha. udtryk (1) undersøge om det pågældende punkt tilhører Mandelbrot mængden, dvs. vi skal undersøge om størrelsen af z_n bliver større end 2. Det kan vi nu ikke gøre helt præcist, for skulle vi gøre det med den her beskrevne metode, ville vi jo få en uendelig løkke lige så snart det punkt vi fik fat i tilhørte Mandelbrot mængden. Så vi må vælge det største antal gange vi vil bruge udtrykket (1) på hvert punkt. Der er delte meninger om hvor stort dette tal bør være; jeg har set tal fra 100-1000 foreslået. Skal man have en chance for at generere et billede på en rimelig tid, skal man ikke vælge et alt for stort tal. For hvert punkt noterer vi os nu, for hvilket n størrelsen af z_n bliver større end 2; bliver $|z_n|$ ikke større end 2 inden n når den før valgte største værdi for dette punkt, noterer vi størsteværdien. Nu mangler vi kun at vælge hvilken farve vi vil tildele de enkelte pixels på skærmen (som vi jo lader svare til hver sit komplekse tal). Hvis vi har 4 farver til rådighed og har valgt 100 som den størst tilladte værdi, kunne vi f.eks. vælge at punkter, hvor $|z_n|$ bliver større end 2:

```
for n mellem 0 og 15 får farve 1
for n mellem 16 og 45 får farve 2
for n mellem 46 og 99 får farve 3
og endelig får dem hvor n = 100 farve 4.
```

Det kunne selvfølgelig gøres på mange andre måder. Faktisk kan man risikere at få valgt intervaller på en måde så billedet bliver helt uinteressant, så det betaler sig at gemme n -værdierne for de enkelte punkter, så man kan eksperimentere med forskellige farve-

valg uden at skulle gennemføre de tidskrævende beregninger igen.

Det er en utrolig detalje rigdom, der på denne måde kan findes i området omkring Mandelbrot mængden. Selv om man nu har fået lavet et billede i en forstørrelse, skal man ikke tro det stopper med det! Hvis man nu vælger et nyt område, der ligger inde i det område man lige har arbejdet med, og gentager proceduren for dette område, så vil man finde nye detaljer og strukturer her.

Hvis du er interesseret i at læse om Mandelbrot mængden mv. kan du f.eks. kikke i:

BYTE fra december 1986, s. 207-210

Scientific American fra august 1985, s. 16-24

Der er også en bog der hedder:

The Beauty of Fractals, skrevet af
H.-O. Peitgen og P.H. Richter, udgivet på
Springer Verlag i 1986

En del af de billeder, der blev vist i den i indledningen omtalte HVÆLV udsendelse, kan iøvrigt genfindes i denne bog.

Lars Gråbæk

Efterlysning

Hermed efterlyses alle, der stadig er interesseret i det famøse CPM+ projekt.

Vi ved, at der i sin tid blev solgt en del kort til dette projekt, og at flere, der købte disse kort, er gået helt væk fra MPS-maskinen. Vi er dog stadig en lille gruppe, der ønsker at fuldføre projektet og tilpasse software'en til de generelle krav VI måtte stille (1 MB ramdisk o.lign.). Det vil sige, at der ikke bliver lavet unødige ting, som VI ikke har brug for.

Da MUG ikke længere er inde i billedet, ønsker vi at samle de få, der stadig kunne have interesse i projektet. Hører DU til denne gruppe, så skriv et postkort med din tilkendegivelse, navn, adresse, telefonnr. og evt. træffetid, så vi kan kontakte dig hvis der er spørgsmål.

Send postkortet til:

Søren Ree
Degnemoose Alle 18
2700 Brønshøj

Mandelbrot program

På medlemsdisketten, der følger med dette nummer af brugerbladet, ligger der 2 programmer til generering af Mandelbrot-mængder. Programmet er skrevet direkte af fra en artikel i PC World, februar 1988.

Programmet er skrevet i Turbo Pascal version 4.0, og kræver adgang til denne kompilator, hvis man selv vil ned og pille i det. Kan man nøjes med at bruge det, som det er, ligger der 2 versioner af programmet på disketten. MANBRT er for alle os, der ikke har en matematik-processor i vores PC'er, mens MANBRT87 kræver en 8087 matematik-processor. Matematik-processoren nedsætter beregningstiden for genereringen af Mandelbrot-mængder til under en femtedel. At det er nødvendigt at lægge 2 forskellige versioner af det samme program på disketten, viser en af svaghederne ved den nye Turbo Pascal compiler. Den mangler mulighed for selv at finde ud af, om der er en matematik-processor i maskinen.

MANBRT er afprøvet på maskiner med CGA, EGA og VGA skærmadapters, og burde også køre på Olivetti og maskiner med Hercules grafik, men jeg har ikke haft mulighed for selv at afprøve det. På disketten ligger der en stribe .BGI-filer. Disse filer er drivere til de forskellige skærmtyper, og de skal ligge i det sub-dir/diskette-lev man starter programmet fra.

Vår man starter programmet, skal man først have beregnet et billede. Det gøres ved at vælge "N)yt billede". Herefter indtastes et filnavn (NB. uden extension) og x-koordinat = -0.75, y-koordinat = 0, horisontal længde = 3 og iterationer = 70. Disse værdier giver et billede af grundmængden. Så begynder programmet beregningerne. For hver 256 punkter skriver det ud, hvor langt det er nået for at man ikke skal tro, at maskinen er gået i stå. Og det er virkelig nødvendigt. Min standard PC'er med 4.77 Mhz, EGA-skærm og uden matematik-processor bruger 9 (skriver ni) timer til at beregne grundmængden! Hvis der er plads på medlemsdisketten, ligger der på disketten en billedfil til grundmængden, men kun til CGA og EGA skærme.

Når så billedet endelig er kommet frem, kan man med et tryk på "F" få en farveskala frem, så man ved hjælp af piletasterne kan bestemme farvesammensætningen i billedet. Når man er tilfreds, trykkes på INS-tasten og billedet gentegnes. Vil man tilbage til den oprindelige farvesammensætning trykkes på ESC.

Hvis man vil beregne et nyt billede, kan man med "Z" få en lille firkant frem på skærmen. Den kan man flytte rundt med piletasterne (det går langtsomt), og gøre den større/mindre med PgUp/PgDw. Når man har fundet det udsnit man vil beregne, trykkes på INS-tasten, og man giver sig til at vente.

Jørgen Petersen

C++

Objektorienteret programmering i C

Bjarne Gertz Pedersen

C++ er en overbygning til det meget anvendte sprog C. Den væsentligste forbedring er, at der i C++ gives fleksible og effektive muligheder til at definere nye typer. Det grundlæggende koncept i C++ er således klassebegrebet, kendt fra Simula67 og Concurrent Pascal. Klassen er en brugerdefineret type, som giver mulighed for at knytte datastrukturer og operationer sammen.

Denne teknik kaldes for data abstraktion. Det fundamentale i abstrakte datatyper er at skille implementationen fra anvendelsen.

Da kun operationerne er tilgængelige, opnås stor sikkerhed og implementeringsdetaljerne kan skjules. Programmet kan derved deles ned i en række enheder, der ligger tæt på den opgave, som skal løses.

Baggrund

C++ er udviklet på Bell Labs., hvor det almindelige C sprog også har sin oprindelse. Der har ikke været tale om et egentligt C++ projekt, men ideen til sproget opstod, da Bjarne Stroustrup skulle skrive nogle hændelsesstyrede simuleringsprogrammer.

Til denne opgave ville Simula67 have været ideel, men pga. effektivitetshensyn blev det ikke valgt. I 1983 blev C++ for første gang installeret uden for Bjarne Stroustrups udviklingsgruppe på Bell Labs. og i dag er sproget det mest anvendte objekt orienterede sprog i USA.

Kompatibilitet

Et væsentligt designkriterium for C++ har været at bibeholde kompatibiliteten med C. Den basale syntaks og semantik er ens for de to sprog.

Dette betyder at:

- de biblioteksrutiner, der er skrevet i C, også kan bruges i C++ programmer. C og C++ kan lænkes sammen.
- de mange C programmører umiddelbart kan anvende C++ og når det bliver nød vendigt med bedre muligheder for typecheck og data abstraktion, kan disse faciliteter tages i brug.

C++ "oversætteren" er ikke en almindelig oversætter i den forstand, at den genererer objektkode, men derimod en translator, der virker som front-end til en almindelig C oversætter, dvs. den genererer C kode, som derefter skal oversættes af en almindelig C oversætter.

Der findes C++ translatorer til en lang række C oversættere og herunder kan jeg blot nævne nogle af de mere kendte:

DOS: - Lattice, Microsoft.

Andre: - VAX C, SUN C, APOLLO, HP-9000, XENIX, UNISOFT.

Desuden findes det kendte PC bibliotek PforCe nu også i en PforCe++ version.

Effektivitet

C++ bibeholder C's effektive muligheder for at kommunikere med hardware. Da de indførte begreber i C++ kun giver translatoren bedre muligheder for kontrol i oversættelsesfasen og da den genererede C kode oversættes af almindelige C-oversættere, vil der ikke gå noget tabt mht. eksekveringshastighed, men derimod vil der kunne vindes meget i udviklingstid, især udvikling af større program-komplekser.

C++ udvidelser

Som allerede antydnet er en række forbedringer indført i forhold til standard C. Den væsentligste er muligheden for at anvende brugerdefinerede typer. Klassen indeholder en privat datastruktur, samt en mængde operationer, der kan operere på denne datastruktur. Et eksempel på en klasse kan være "ostream" fra I/O biblioteket. Den indeholder en buffer, hvor tegn til standard output kan gemmes, samt de operationer, der kan gemme tegnene i bufferen. Erklæringen ser således ud:

```
class ostream{
    streambuf * buf;
    int state;
public
    void put (char*);
    void put (long);
    void put (double);
};
```

buf og state er de private datastrukturer, mens put operationerne er interfacen for programmer, der ønsker at skrive til standard output. En erklæring af en klasse beskriver blot typen af klassen og derfor er det nødvendigt at definere en variabel af klassen. I det følgende vil en variabel blive benævnt som et objekt. Helt præcist, i følge terminologien inden for objekt orienteret programmering, er det kun datastrukturerne, der er objekter, mens operationerne er meddelelser, som sendes til et objekt af en klasse.

fortsættes side 23

Volume PC-SIG 460

ALBERTA AGRICULTURAL PROGRAMS

This seems to be another useful collection of decision aids for farm management. Basica is required.

AUTOEXEC	BAT	Batch file basica hello
BESTCROP	BAS	Breakeven prices & yields - march 1982
BFP	BAS	Breakeven feeder prices - july 1984
CARCOST	BAS	Car or truck costing program - january 1984
CFP	BAS	Comparative feed pricing - june 1984
COWCRAB	BAS	Cow calf risk analysis budget - july 1983
CRCROP	BAS	Consensus research crops model - july 1984
DIET	BAS	Diet detective jr. - march 1983
DOLLAR	BAS	Dollar detective - personal budgeting - march 1983
HAYCRAB	BAS	Hay cost returns and analysis - march 1982
HELLO	BAS	Alberta agriculture menu program - august 1984
INSTALL	BAT	Batch file to add system and basica to disk
LANDPUR	BAS	Land purchase decisions - february 1983
LOAN	BAS	Loan calculator - august 1984
RATION	BAS	Ration balancing - may 1984
README	DOC	Author supplied documentation file

Volume PC-SIG 461

RIDGETOWN COLLEGE PROGRAMS

Here is another fine collection of farm management tools submitted from Ridgetown college of Ontario. Basica is required.

AVERAGE	DAT	Unkown ??? data file for this disk
CONV	BAS	Converts programs from exe or com to bas
LUMBER	BAS	Lumber requirements for building
LUMBER	DOC	Documentation for LUMBER.BAS
METRICON	BAS	Metric conversion program - source
METRICON	EXE	Metric conversion program - compiled
QUIZ	BAS	Do you need a farm computer quiz - source
QUIZ	EXE	Do you need a farm computer quiz - compiled
README	DOC	Author supplied user documentation file
SHRINK	EXE	Shrinks basic files prior to compilation
SILENCE	COM	Silences your computer speaker
TREEFIX	BAS	Fixes output from TREE.COM
TREEFIX	BAT	Batch file to produce useful form of TREE.COM
USLE	BAS	Universal soil loss equation
USLE	DOC	Documentation for USLE.BAS
WOODHEAT	BAS	Economics of wood heating
WOODHEAT	TXT	Documentation for WOODHEAT.BAS

A series of programs that tracks your income and expenses, prints many types of reports regarding what you've earned and where you've spent, cross-referenced many ways, draws bar charts on income/expenses, keeps an orderly check register/deposit record, etc. Very useful system with extensive, detailed documentation for the inexperienced user. The programs on disk \$1 are used to enter check data, examine files, correct mistakes, add files together, set-up income/expense categories & list of checking accounts. The user manual is contained on this disk also.

CK	EXE	Main program
CKACCTS	EXE	Sub-program : set-up your list of accounts
CKADD	EXE	Sub-program : adds data files together
CKEXSET	EXE	Sub-program : set-up income and expense categories
CKEXTOT	EXE	Sub-program : totals expenses
CKINPUT	EXE	Sub-program : enter/change at check information
CKLOOK	EXE	Sub-program : look at check information
CKMSG		Text file containing messages used by program files
CKREAD	ME	Documentation/user manual (wordstar document) 28 pgs *

A series of programs that tracks your income and expenses, prints many types of reports regarding what you've earned and where you've spent, cross-referenced many ways, draws bar charts on income/expenses, keeps an orderly check register/deposit record, etc. Very useful system with extensive, detailed documentation for the inexperienced user. The programs on disk \$2 are used to do year-to-date reports and bar charts.

CK	EXE	Main program
CKAMTD	EXE	Sub-program : prepare report sorted by amount spent
CKCAT	EXE	Sub-program : prepare report sorted by expense category
CKCHRT	EXE	Sub-program : prepare a chart
CKDATE	EXE	Sub-program : prepare report sorted by date of expense
CKINST	EXE	Sub-program : install system
CKNAME	EXE	Sub-program : prepare report sorted by account name
CKNUMBR	EXE	Sub-program : prepare report sorted by check number
CKSORT	EXE	Sub-program : sort routine
SORT	EXE	Sub-program : used bu CKSORT.EXE program

PBASE is a programmable relational database management system designed for programmers. It provides facilities for querying and manipulating data, as well as report writing. Unlike most database managers, PBASE represents all data as a table. Data may be accessed & revised using interactive commands or through user-written application programs. PBASE is based on the structured query language (sql).

EDITFORM	PRG	A pbase program to edit forms
EXPENSE	PRG	Sample pbase program to track business expenses
INVENTORY	PRG	Test program - inventory control
MAILLIST	PRG	Sample program - keep mailing lists, etc.
PB	EXE	Main pbase program
PBASE1	REF	Text file containing the documentation/user manual
README		Text file containing 'how to get started', etc.
REALTY	PRG	Test program - real estate listing property info
WINE	PRG	Sample program showing simple sql commands - wine cellar

A genealogy program for genealogists to simplify their work. It is designed for the organized compilation of personal genealogical information.

COPYPROG	BAT	Batch program to copy the program files
FGRSH	INS	Messages file
FT	EXE	The main family ties program
FTCOLOR	DAT	Data file containing color settings
FTDEMO	EXE	Demo/tutorial program
FTINSTAL	DAT	Printer data used by the installation program
FTINSTAL	EXE	Installation program for family ties
LICENCE	DOC	Information on the licence for the program
MANUAL		The user manual/documentation text file (20 pages)
NAMES	DA1	Database constants
NAMES	NTS	Blank notes file
PRINT	CTR	Printer control file
README		Information regarding the files on this diskette
README	BAT	Batch file to display the readme file
REGISTRN	FRM	Text file of the registration form
UPDATE	DOC	Information on how to update early versions
UPDATE	BAT	Batch program to update early versions of family ties
USER-NO	EXE	Program to change the user number encoded in a family ties Master disk. For use when registering as a user.
QEURY	COM	

This is your "program diskette". It contains all of CPA-ledger's computer programs. Basica is required.

ACTDTL	BAS	This program processes the g/l files from your data disk
BALSHT	BAS	This program allows the user to generate the balance sheet
CHKREG	BAS	This program prints transactions that inc/dec bank accts
DAILY	BAS	This program handles most daily operations/transactions
GENLGR	BAS	This program is used for establishing a new general ledger
GLACTS	BAS	This program prints a chart of general ledger accounts
NEWACT	BAS	This program allows the user to add new accounts to g/l
OPNBAL	BAS	This program opens and posts balances to the g/l
PANDL	BAS	This program handles primary & contra accts & income statements
POST	BAS	This program allows the user to post entries
PSTCLS	BAS	This program will prepare a post-closing trial balance
READ	ME	Author supplied documentation file
SELTRA	BAS	This program selects input ranges and dates
SETUP	BAS	This is the user's initial "setup" program
STOACT	BAS	This program sets up and stores the accounts
TRIBAL	BAS	This is the trial balance program

This diskette contains CPA-ledger's "practice diskette". Basica is required.

GENJNL	FIL	This is the general journal file
GENLEG	FIL	This is the general ledger file
INVPRO	FIL	Part of ledger program
PASACT	FIL	Part of ledger program
READ	ME	Author supplied documentation file
TEST1	FIL	Test1 is a practice file
TEST2	FIL	Test2 is a practice file

This diskette contains CPA-ledger's user's manual, which contains over 150 pages. Appendix d contains some comments, about accounting systems, that you might find helpful. Appendix E contains some forms that may be useful in summarizing your accounting entries prior to entering them into CPA-ledger.

APNDX-A TXT Using CPA-ledger with two floppy disk drives
APNDX-B TXT Using CPA-ledger with fixed disk drives
APNDX-C. TXT Preparing your data diskette and storage capacities
APNDX-D TXT Some comments about an accounting and bookkeeping system
APNDX-E TXT Contains forms for summarizing data before entering into CPA-ledger

CHP1 TXT CPA-ledger overview
CHP2 TXT Standard messages and reminders
CHP3-5 TXT Setup menus - options 1,2 and 3
CHP6-7 TXT Daily menus - options 1 and 2
CHP8 TXT Daily menu - option 3
CHP9-11 TXT Daily menus - options 4,5 and 6
CHP12-15 TXT Daily menus - options 7,8,9, 10 plus setup menu - option 4
PREFACE TXT Introduction, table of contents, disclaimer etc.
READ ME Author supplied documentation file

A series of programs that prepares itemized invoices and bills, reports items billed/credited, ageing reports for clients, audit trail of billing entries, and summary reports. Very flexible, many uses.

BILL EXE Main billing program
BILLADD EXE Sub-program : enter/change/delete/look at name/addresses
BILLIN EXE Sub-program : enter billing data
BILLINIT EXE Sub-program : initialize system
BILLMERG EXE Sub-program : merge data from several small files
BILLREAD ME Documentation/user manual (wordstar document) #U20 pgs)
BILLS Data file - serial number of next bill
BILLSET EXE Sub-program : set/change/delete/look at codes/rates
BILLSORT EXE Sub-program : sorts billing entries
BILLSUB EXE Sub-program : enter/change/delete/look at subtotals
ORD Blank order data file

A series of programs that prepares itemized invoices and bills, reports items billed/credited, ageing reports for clients, audit trail of billing entries, and summary reports. Very flexible, many uses.

BILL	EXE	Main billing program
BILLAGE	EXE	Sub-program : produces ageing report
BILLDATE	EXE	Sub-program : produces report of billings by date
BILLRPT	EXE	Sub-program : prepares bills
BILLRPTP	EXE	Sub-program : printed report of bills
BILLS		Data file - serial numbers
BILLSORT	EXE	Sub-program : sorts billing entries
BILLTRL	EXE	Sub-program : audit trail
MBINST	EXE	Sub-program : part of installing system
SORT	EXE	Sub-program : used by billsort sub-program

This diskette contains the complete set of present programs (version 5.1), and associated utilities. It also has a set of pictures (.Pic) that can be used to provide a demonstration. To start a self-running demo enter: present demo *

The system is documented fairly extensively in PRESENT.DOC. Print it out and browse through it to familiarize yourself with all features. Requires 128k of memory and color card

AND	PIC	Present slide picture file
BIGSNOOP	PIC	Present slide picture file
BLACK	PIC	Present slide picture file
CAMERA	COM	Memory resident utility takes pictures of screen images
CAMERA4	PIC	Present slide picture file
CAMERAF	PIC	Present slide picture file
CHARLIE3	PIC	Present slide picture file
CLOSE	PIC	Present slide picture file
COMPRESS	COM	Converts basic bsave files to present's internal format
DEMO		This program is a demonstration of the features of present
DISOLVEN	PIC	Present slide picture file
EXPAND	COM	Converts present files to basic bsave format
EXPAND	PIC	Present slide picture file
FADEIN	PIC	Present slide picture file
FEATURES	PIC	Present slide picture file
FTCAM	PIC	Present slide picture file
G	PIC	Present slide picture file
GA	PIC	Present slide picture file
GAR	PIC	Present slide picture file
GARF	PIC	Present slide picture file

GARFI	PIC	Present slide picture file
GARFIE	PIC	Present slide picture file
GARFIEL	PIC	Present slide picture file
GARFIELD	PIC	Present slide picture file
GRAPHHELP	PIC	Graphics help file
HAS	PIC	Present slide picture file
HAS12	PIC	Present slide picture file
LETSSEE1	PIC	Present slide picture file
LETSSEE2	PIC	Present slide picture file
MERGE	PIC	Present slide picture file
OPEN	PIC	Present slide picture file
P	PIC	Present slide picture file
PIC2RAS	COM	Picture to raster utility program
PR	PIC	Present slide picture file
PRE	PIC	Present slide picture file
PREPARE	COM	Use this program to "prepare" slides for presentation
PREPARE	HLP	This is prepare's help file
PREPAREP	PIC	Present slide picture file
PRES	PIC	Present slide picture file
PRESE	PIC	Present slide picture file
PRESEN	PIC	Present slide picture file
PRESENT	COM	The main executable file
PRESENT	DOC	Author supplied documentation file
PRESENT	HLP	Help file for present program
PRESENT	PIC	Present slide picture file
PRESENT2	PIC	Present slide picture file
PRESENTP	PIC	Present slide picture file
PROGS	PIC	Present slide picture file
PULLDOWN	PIC	Present slide picture file
PULLEFT	PIC	Present slide picture file
PULLUP	PIC	Present slide picture file
PULRIGHT	PIC	Present slide picture file
RAS2PIC	COM	Raster to picture utility program
README	1ST	User supplied documentation file
REPLACE	PIC	Present slide picture file
SLIDE4	PIC	Present slide picture file
SNOOPY	PIC	Present slide picture file
SNOOPY04	PIC	Present slide picture file
SQUEEZE	PIC	Present slide picture file
SWEEP	PIC	Present slide picture file
TEXTHELP	PIC	Text help file
TREE	PIC	Present slide picture file
UTILITY	PIC	Present slide picture file

The minimum hardware configuration for using these programs is an ibm-pc or compatible computer with 128 k of memory, 2 ds dd disk drives, and an ibm, epson or other compatible printer.

A EXE Information on how to use to *.BAT files to backup data
ACCTREC BAS Processes accounts receivables
AUTOEXEC BAT Request the date then loads basica bkpg /f:5
BASICA COM Ibm pc basic interpreter program
BKPG BAK Backup file of bookkeeping
BKPG BAS This is the bookkeeping program
BKPG DOC Author supplied documentation file
C BAT COPY PAYEE.FIL, CATEGORY.FIL, INCOME.FIL to B:
CHKDSK A: & B:
CHECKING BAS Maintains the checking account
CHKDSK COM Ibm pc chkdisk program, disk check utility
EXPENSE BAS Keeps track of expenses
F BAT Creates/formats two new disks, for bookkeeping fiscal year
FORMAT COM Ibm pc format program, formats floppy or hard disks
G BAT Copies totals to new backup bookkeeping disk, runs BKPG.BAS
INCOME BAS Keeps track of income
L BAT Copies CHECKING.ACC to new backup bookkeeping disk, runs
BKPG.BAS
N BAT Creates two disks then loads basica and runs payroll
program
PAYROLL BAS Processes the payroll
R BAT Creates two disks then loads basica and runs a / r program
S BAT Format 6 disks by responding "y" first 5 times, for
startup
SHELLSORT BAS This is a shell sort program for sorting files
STARTUP BAT Author supplied user startup procedure batch file
X BAT Makes new backup copy of payroll file disk on drive b:

Trivial towers is a trivia game that tests the users knowlege on a variety of different subjects. It is a game that 2 to 4 people can play at a time. By answering questions correctly, you build "TOWERS." When you complete all 4 towers, that are 6 blocks high, you win the game. There are special blocks on the gameboard, that when landed on, enable the user to choose any topic of his/her choice.

Trivial towers (V1.0)
TTENTER BAS Program that enables the user to write customized questions.
TTOWERS BAS Main program
TTRULES BAS Rules for TTOWERS.BAS and directions on how to customize.
TTNOTE BAS A letter from the authors of the program.
TTSHARE DOC A welcome letter from the authors giving definitions.

This disk is all the questions and answers for the trivial towers program.

Trivial towers (V1.0)
INCOMMON Question and answer file
CAPITALS Question and answer file
VANITY Question and answer file
COMPUTER S Question and answer file
TVFILM Question and answer file
WRITERS Question and answer file

Monopoly P.C. is a game developed to resemble the famous board game by Parker Brothers. This program follows the rules of the board game closely and allows one player to play against the computer, which rolls the dice moves the pieces on the board and updates the screen so that the player can follow along. There is also a graphics screen which displays the game board.

Trivia tune and its associated data files comprise a software package that allows players to hear music from the IBM's speaker and then attempt to guess the song's title. Trivia tune then allows the players to answer trivia questions about musical history once a correct song has been guessed.

Both programs require a minimum of 128k of memory to run. The graphics card for the Monopoly P.C. Game is optional.

----- Monopoly P.C.

MONOPOLY	EXE	Main program
MONOP	SCR	Data file to create graphic screen image
MONOPM	SCR	Data file to create graphic screen image
MONOPOLY	LCL	Modified "local rule" option text file
MONOPOLY	HLP	Brief help file displayed when requested during game
-----		Tune trivia (V1.0)
README		Brief information about authors and program
MANUAL		Program instruction manual (22k) set for 8.5" x 11" paper
TUNETRIV	EXE	Main program for tune trivia
SET0??		Data files holding musical notes for TUNETRIV.EXE (22)
TRIV0??		Data files holding trivia questions for TUNETRIV.EXE (21)
SIGSOFT		Signature data file (company name only)

These games were gathered from previous pc-sig disks to provide a good representation of high quality games for the new user. In answer to complaints that most of the best games require color screens, only 3-demon and spacevad out of the seven included games need a color graphics card.

GO BAT File that prints help messages
README1
README2
README3
README4
PACKGAL COM Multiscreen version of ms. Pacman with a few twists.
POKER EXE The school of implemented casino probability theory.
BOOGERS! BAS A maze game that requires destruction of blockades.
CASTLE EXE An adventure game with clever animation and graphics,
CASTLE RAN It has elements of role-playing and strategy games.
BUGS EXE A good version of the classic arcade game centipede
3-DEMON EXE Three-d view of pacman-like game from inside the maze.
SPACEVAD EXE Excellent version of space invaders, great graphics
3-DEMON HI1 Records the high score of 3-demon game for posterity.

Namegram is a program that takes an input word, usually a name, and compares it to a set of compressed word files. In doing this, it can generate a set of anagrams of the word or name. Break down is a strange, but interesting program. It has almost no practical uses, except perhaps for entertainment, but this does not limit its value. This algorithm takes an input text file (i tried about two paragraphs of an old english paper) and transforms the source text into a bunch of sentences that make little to no sense, but somehow they do!?

----- Name gram
NAMESMAL COM Main program
MASHED 02 Compressed word files used by namegram
MASHED 03 Compressed word files used by namegram
MASHED 04 Compressed word files used by namegram
MASHED 05 Compressed word files used by namegram
MASHED 06 Compressed word files used by namegram
READNAME COM Screen driven documentation program for namegram (33k)
----- Break down
BREKDOWN COM Main program
BREKDOWN PAS Pascal source code for brekdown program
BREKDOWN DOC Short documentation for brekdown (3 pages)
BREKDOWN XMP Some examples of text generated by brekdown (3 pages)
BREK2 TXT Associated text file used by brekdown
ROMEO 05T Order-5 breakdown of the "window scene" from romeo & juliet
ROMEO 08T Order-8 breakdown of same, cleaned up and punctuated.
REAGAN OUT Order-8 breakdown of the '85 state of the union address
----- Phone word
FONEWORD COM Main program
FONEWORD DOC Documentation of foneword program
WORDFONE COM Reverses foneword program
----- Bonus files: created using my pianoman program
WILDWEST COM "the wild west is where i wanna be" by tom lehrer
WILLTELL COM "the william tell overture" by G.A. Rossini
INVENTN4 COM Two-part invention #4 in d-minor, by J.S. Bach
MNTPYTHN COM Theme from monty python's flying circus, by J.P. Sousa

C++ notationen understøtter ikke helt tankegangen i objekt orienteret programmering, hvor der altid sendes meddelelser til objekter, så derfor vil jeg i det følgende ikke anvende ordet meddelelse for en operation eller funktion, men dog bibeholde benævnelsen objekt for en variabel af en datastruktur.

I erklæringen af klassen fremgår det, at der er 3 operationer med samme navn, put(), men med forskellige parameter overførsler. Det er tilladt i C++ at anvende samme navn for forskellige operationer. Dette er specielt anvendeligt, når man har en række operationer, der udfører næsten den samme ting, men som her på forskellige datatyper. Dette kaldes for "overloaded function names" og oversætteren skelner mellem funktionerne på baggrund af parameter overførslen.

Hierarki af klasser

Et af de bedre redskaber til at organisere komplekse problemstillinger er at anvende hierarkiske strukturer. Det gælder om at anvende en træstruktur, hvor objektet i roden indeholder de mest generelle ting og jo længere man kommer ud af forgreningerne jo mere specialiseret bliver objekterne.

Denne form for opbygning er en af de fundamentale ting, der skal understøttes af objekt orienterede sprog. I C++ gøres det ved at klasser kan "arve" hinandens egenskaber, på engelsk benævnes det som "Inheritance".

Hvis man skulle beskrive eller organisere musikinstrumenter kunne det gøres ved at have en basisklasse "Baseclass", hvor de generelle egenskaber, der gælder for alle musikinstrumenter er defineret.

Forgreningerne kunne derefter være strengeinstrumenter, blæsere, slagtojsinstrumenter m.m. De klasser vil, foruden deres egne specielle egenskaber, stadig også være omfattet af basisklassens egenskaber.

Disse klasser kaldes afledte klasser "Derived Classes". Klassen af strengeinstrumenter ville så igen kunne opdeles i en række afledte klasser, nemlig strygere, guitarer m.m. og på denne måde kunne der fortsættes.

Basisklassen for strygere ville nu være strengeinstrumenter.

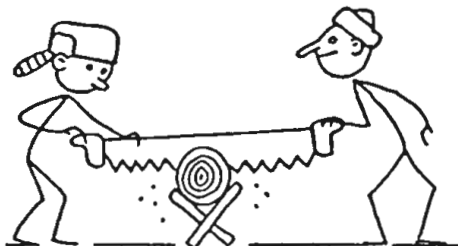
Hvis dette føres videre til C++ betyder det, at hvis et objekt af en klasse bedes om at udføre en operation vil den også kunne bruge de operationer, der er defineret i dens basisklasse, hvis det er nødvendigt for at udføre operationen.

Dette betyder igen, at afledte klasser kun bør tilføje operationer eller egenskaber til hierarkiet.

Der findes langt flere begreber i C++, men det vil være for omfattende at komme nærmere ind på disse i denne artikel. Jeg kan blot nævne nogle af dem, f.eks. virtuelle funktioner, garanteret initialisering af klassesdata, dynamisk typebinding, inline substituering af funktioner, operatorer til lagerstyring, operator overload og default parametre.

Reprint af artikel skrevet i forb. med Teknologisk Institut / Automatiserings-teknik's kursus 6-8 April & 6-8 September 1988. Yderligere oplysninger om C++ og kursus kan fås ved henv. til Bjarne Gertz Pedersen, (02) 99 66 11 lokal 142

(VJ)



Cee - Saw

Bulletin Board

PASCAL KONFERENCE

From: Sveinar Aanes
To: All
Subject: IOResult TP 4

05-04-88 18:27

Problemer med IOResult (TP 4)

Nedenforstaaende del av et program som gikk utmerket under TP 3.0 har gitt meg graa haar naar jeg kjoerer det under TP 4.0.

Denne delen av programmet skal lese inn en ascii-fil bestaaende av bare tall (integer).

For aa teste programmet oppgir jeg en ikke eksisterende fil. Foer-

ste gangen reagerer programmet riktig, mens andre gangen gaar det til linje 12 og angir error-code 106: Invalid numeric format.

```
1) READ(filnavn);
2) ASSIGN(innfil,filnavn);
3) æSI-å
4) RESET(innfil);
5) æSI+å
6) IF IORESULT = 0 THEN
7) BEGIN
8) funnet:=TRUE;
9) WHILE NOT EOF(innfil) DO
10) BEGIN
11) lengde:=SUCC(lengde);
12) READLN(innfil,tall);
13) adrÆlengdeÅ := tall;
14) END;
15) GOTOXY(13,8);
16) WRITE(lengde-1,' ADRESSER ER LEST INN');
17) END
18) ELSE
19) BEGIN
20) funnet:= FALSE; slutt:=FALSE;
21) GOTOXY(12,10);WRITE('FINNER IKKE ',filnavn,');
```

Etter en del forsoek har jeg kommet fram til foelgende endringer:
linje 1 endres til: READLN(filnavn);
" 6 " " : IF IORESULT < 1 THEN

Jeg har ikke funnet noe i manualen som tilsier disse endringene, men programmet fungerer etter hensikten.

Finnes det en forklaring ? ? ?

From: Kaare Rasmussen
To: Sveinar Aanes
Subject: IOResult TP 4

06-04-88 09:14

Hej Sveinar

Nå, jeg troede, jeg var den eneste, der var kommet ud for det. Det er ganske rigtigt read, der er problemet, linien med ioresult tror jeg godt, du kan rette tilbage (ioresult returnerer et word). Når du kommer tilbage til den read, står der stadigvaak cr/lf i bufferen. Det vil read(string) ikke acceptere og returnerer tom streng og fjerner stadigvæk ikke cr/lf. Und so weiter ... Det er ikke dokumenteret, fordi (mener jeg) det er en fejl. Men man kan f.eks. se det ved at lave en read(ch) og kigge paa den.

Med venlig hilsen Kaare

From: Peter Rasmussen
To: Sveinar Aanes
Subject: IOResult TP 4

06-04-88 20:26

Jeg har også haft problemer med TP4.0 I/O check og IOResult. Jeg har fundet ud af, at hvis du lukker en fil som ikke er åben, så går det helt agurk. Der sker det sjove, at den f.eks. finder på at lukke for con (skærm og tastaturet) og dette medfører at den næste read eller write giver 'file not open' !!! Et eksempel :

```
assign(infile, 'test.xxx');
æSI-å reset(infile); æSI+å
IF ioresult=0
THEN BEGIN
  "
  readln(infile, buffer);
  "
END;
"
æSI-å close(infile); æSI-å
"
"
write('slut');
```

Hvis test.xxx ikke findes på disken, vil programet ovenfor 'lukke' for filen og dette medfører en IOError. Denne IOError fører videre til write sætningen til skærmen..... Så derfor: Luk kun filer man ved er åbne !!!!!

En anden ting jeg fandt ud af var, at hvis man skrev til printeren eller til en com port (også åbnet med assign/reset) kunne dette give runtime error, hvis enheden ikke var klar (on-line etc.), på den anden write access. En måde at klare dette på er at checke IOresult efter hver write ! F.eks.:

```
æSI-å
writeln(printer, 'Line 1');
dummy:=ioresult;
writeln(printer, 'Line 2');
dummy:=ioresult;
writeln(printer, 'Line 3');
dummy:=ioresult;
æSI+å
```

Jeg ved ikke om du kan bruge det her til noget, og jeg er ikke 100% sikker på rigtigheden af det jeg har skrevet. Det er nogle erfaringer jeg har gjort i et større program, og jeg har ikke efterprøvet det i et testprogram for at se om det var den fulde sandhed. Man faktum er at det hjalp i mit 'store' program. OK?

<Peter>

From: Peter Rasmussen
To: Sveinar Aanes
Subject: IOResult TP 4

06-04-88 20:35

Hej igen.

Hvor closer du filen??? Hvis du gør det efter linje 21 går det galt! Prøv at inføje en linje mellem 16 og 17 : close(innfil); Husk at fjerne samme linje længere nede i programet (Hvis du ikke helt har glemt den linje :-).)

Virker det ?.....

<Peter>

From: Torben Paving
To: Peter Rasmussen
Subject: IOResult TP 4

07-04-88 23:17

Hej Peter!

Det fænomen du beskriver, er tydeligt doukumenteret i min manual pg. 438, hvor der bl.a. står:

* If an I/O error occurs and I/O checking is off, all subsequent I/O operations are ignored until a call is made to IOResult. *

Compileren opfører sig derfor som den bør. Du kan da heller ikke med god samvittighed slå I/O checking off og derefter bare køre derudaf uden at kontrollere nogetsomhelst, vel?

Hilsner Torben

From: Peter Rasmussen
To: Torben Paving
Subject: IOResult TP 4

09-04-88 18:24

Jep, men hvis jeg ikke husker helt galt (Du må ikke hænge mig op på det) så fik jeg en runtime error på anden write, selv om jeg havde æ\$I-å sat.....

Compileren opfører sig derfor som den bør. Du kan da heller ikke med god samvittighed slå I/O checking off og derefter bare køre derudaf uden at kontrollere nogetsomhelst, vel?

Nej da..... Jeg ville bare kontrollere den sidste write sætning !

MUG 1988:2

- 27 -

Har du prøvet at lukke en fil der ikke var åben??? Ok, det er måske ikke noget pænt program der prøver den slags, og jeg gør det da heller ikke mere.

<Peter>

From: Sveinar Aanes
To: Peter Rasmussen 10-04-88 18:34
Subject: IOResult TP 4

Takk for svaret - det er hyggelig naar man ikke er alene om problemene.

Hvis du ser paa min feorste melding, saa oppstaar problemet naar jeg oppgir et filnavn som ikke finnes paa disken. Foerste gangen reagerer programmet riktig, men andre gangen aborterer programmet med Error-code:106.

Da jeg kjoerte dette programmet gjennom UPGRADE.EXE fikk jeg en melding ved IOResult som sa:

IOResult now returns different values
corresponding to DOS error codes.

Men saa langt jeg har kunnet konstatere er det de samme IO-codene som TP 3.0 brukte.

Selvom jeg har funnet en loesning paa problemet, savner jeg en forklaring - om det finnes.

Angaaende CLOSE <filename>: Det er vel ikke noedvendig aa lukke en fil som bare leses fra? Min erfaring tilsier at det ikke er noedvendig.

From: Peter Rasmussen
To: Sveinar Aanes 12-04-88 18:40
Subject: IOResult TP 4

Nødvendigt og nødvendig, det er vel som man tager det. Det er rigtigt at man SKAL close filer når man skriver til dem, da disk bufferen først tømmes ved close og directoryet opdateres. Efter min mening skal man *ALTID* close filerne når man er færdig med dem, lige meget om man læser eller skriver til dem.

<Peter>

From: Torben Paving
To: Peter Rasmussen
Subject: IOResult TP 4

13-04-88 21:18

Hej Peter!

Har du prøvet at lukke en fil der ikke var åben???

Jo, det er jeg sikker på at jeg har, men så har jeg normalt I/O checking på og så får jeg fejl *103 File not open* og så bliver det rettet! Imidlertid lavede jeg en lille test for nu at være sikker på, hvordan I/O checking fungerede.

Eksempel 1:

```
1: close(testfile);  
2: writeln('Jeg vil majet heller sælle bananer!');
```

Her får jeg fejl 103 på statement 1 og det er vel, som det skal være.

Eksempel 2:

```
1: æ$I-å  
2: close(testfile);  
3: æ$I+å  
4: writeln('Jeg vil majet heller sælle bananer!');
```

Her får jeg først fejl 103 på statement 4. Det må skyldes at IOresult stadig har værdien fra statement 2, og at værdien bliver testet af IOchecking i statement 4.

Eksempel 4:

```
1: æ$I-å  
2: close(testfile);  
3: æ$I+å  
4: if ioresult = 0 then;  
5: writeln('Jeg vil majet heller sælle bananer!');
```

Dette eksempel fungerer perfekt, selvom programmet udadtil er identisk med eksempel 2. IOresult bliver blot resat i statement 4, hvorefter statement 5 ikke mere giver fejl.

Det korte af det lange er altså, hvis du slår I/O checking fra, SKAL du resette IOResult ved at teste eller assigne fra den. Man kan naturligvis med nogen ret diskutere, hvorvidt compileren selv skulle resette IOResult før hver IO-operation, men man har sandsynligvis valgt kun at resette IOResult, når den bliver refereret og kun at sætte den ved fejl, pga performance.

Hilsner Torben

From: Peter Rasmussen
To: Torben Paving
Subject: IOResult TP 4

14-04-88 16:23

Ja Torben, du har helt ret. Jeg bemærkede bare "fejlen" ved at et af mine TP3.0 programmer gav I/O error i TP4.0. Og hvorfor i en write til skærmen ?.....

Jeg mener dog at det er misvisende at melde fejl der hvor den gør, men det er selvfølgelig en lille ubetydelig "fejl" når man kender den.

<Peter>

From: Erik Nielsen
To: Peter Rasmussen
Subject: Re: IOResult TP 4

19-04-88 13:14

Vedrørende snakken om at lukke eller ikke lukke filer:

Når DOS terminerer lukker den alle filehandles, den kørende proces havde åben. Det vil sige, at såfremt ens applikation anvender file handles er det ligegyldigt om en fil lukkes eller ej, når man når til END. klarer DOS resten. Kun i de tilfælde, hvor man roder sådan oppe under DOS' skærter at hun hænger og man må boote, er der fare for datatab fra ikke-flushede buffers, og det har jo ikke noget med lukning af filer at gøre.

Helt anderledes forholder det sig hvis man benytter de antikverede File Controle Blocks (FCB). Her er der ingen automatik og ingen nåde.

Når f.ex. Database Toolboxen kan miste indexeringsdata pga. manglende lukning skyldes det, at indexer først opdateres endeligt med diskskrivning når filen lukkes, men det skyldes altså, at data ikke er overgivet til DOS' bufferering, ikke at buffers ikke flushes.

Noget helt andet er, at det hører ind under dårlig programmerings-skik ikke at rydde op efter sig.

Erik

Adresser

SOFTWARE OG DISKETTER

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

PC-volumes bestilles ved PC-bibliotekaren.

40 spor dobbeltsidet	pr 25 stk.	150,- kr.
40 spor dobbeltsidet, flere farver	pr 10 stk. TILBUD	100,- kr.
Ved forsendelse pålægges ekspeditionsgebyr		20,- kr.

Volume fra biblioteket (excl. disk)	pr. 1 stk.	30,- kr.
Volume fra biblioteket (incl. disk)	pr. 1 stk	50,- kr.

BESTYRELSEN

Formand:

Mikkel Moulvad
Sofiegade 24
1418 København K
Tlf. 01 57 18 92
(hverd. 17 - 19)

Kasserer

Jørgen Petersen
Sofiegade 24, 2.
1418 København K
Tlf. 01 54 91 76
(man - tor: 20 - 21)

Sekretær:

Mogens Kruse
Digterparken 68
2750 Ballerup
Tlf. 02 97 38 38

Disketteredaktør:

Carl Skovgaard Larsen
Pærevangen 19, 1h
2765 Smørum
Tlf. 02 66 11 27
(hverd. 19 - 21)

Redaktør:

Leif Olsen
Kildestrædet 46
2740 Skovlunde
Tlf. 02 94 98 20

Mødearrangør:

Lars Gråbæk
Esbern Snares Gade 6
1725 København V
Tlf. 01 23 92 36

Fælleskøb:

Tage Micheelsen
Slotsherrensvej 48, 2
2720 Vanløse
Tlf. 01 79 11 68

PC-biblioteket

Svend Mailand
Borgbjergsvej 46, st tv
2450 København SV
Tlf. 01 21 64 35
(man - tor: 20 - 21)
Giro 7 49 91 40

CP/M-biblioteket

Jørgen Petersen
Sofiegade 24, 2
1418 København K
Giro 5 68 65 12

BULLETIN BOARD

Tlf. 02 11 77 29
Åbent hele døgnet
300, 1200/75, 1200,
2400 bits/sec, 8 bit,
1 stop, ingen paritet

