

MTX *User-Club Deutschland*

Info 38
04.08.1990

Zweck: Zusammentragen und Austausch von Tips & Tricks u.s.w., Hilfestellung bei allen möglichen Problemen, Aufbau einer Programmbibliothek und Basteln von Hardware-Erweiterungen.

Programme (nur Selbstgeschriebenes): Tausch von kurzen und einfachen Routinen. Gute Programme (mit Dokumentation) können über den Club an alle Mitglieder verkauft werden. Wer solche Programme an uns schickt erhält ggf. Verbesserungshinweise und eine Besprechung im Info.

Mitglied kann jeder werden! Keine Beitragsgebühr! Anmeldung kostet DM 1.-.

Verpflichtungen: Einsendung unseres Anmeldeformulars.

Bitte: Einsendung von Tips & Tricks, Fragen, Antworten, kurzen Routinen, Programmen, Beiträgen zum Info, Hinweisen auf preiswerte Hard- und Software, und was noch so zusammenkommt und andere interessieren könnte.

Club-Info, unser Blatt, verschicken wir ca. 8-wöchentlich. Inhalt ist alles was uns über den MTX/FDX (ohne Copyright) in die Hände fällt. Es kostet nicht über DM 12.- je Exemplar. Jeder kann dazu Beiträge liefern und hier gratis Kleinanzeigen veröffentlichen.

Kosten: Wir berechnen ausschließlich Selbstkosten und verschicken nichts, wenn Ihr persönliches Guthaben nicht reicht! (s.u.)
Schüler, Studenten, Auszubildende, Grundwehrdienstleistende, Rentner und Arbeitslose erhalten einen Nachlaß von 40% auf die zukünftigen Infos nach Einsendung einer entsprechenden Bescheinigung für deren Gültigkeitszeitraum.

Geld/Konto: Für jedes Mitglied führt Herbert zur Nedden ein Konto, von dem die jeweils entstehenden Kosten abgehen. Der Kontostand wird bei jeder Sendung mitgeteilt (er steht über der Anschrift) und kann selbstverständlich jederzeit erfragt werden! Wir verschicken nur gegen Vorkasse!

Einzahlungen bitte auf's Club-Konto: (oder V-Scheck)
(Absender! incl Name und Anschrift bitte nicht vergessen!)
Postgiroamt Hamburg, BLZ 200 100 20,
Herbert zur Nedden, Sonderkonto C, Nr. 3480 00-200

Kontaktadressen:

Herbert zur Nedden	Christian Löhrmann	Hans Gras
Sonnenau 2	Grevenbleck 24	Statenhoek 49
2000 Hamburg 76	3005 Hemmingen 1	NL 1506 VM Zaandam
(040) 200 87 04	(0511) 41 78 77	(0031-75) 17 49 91

Telefon-Sprechzeiten

Herbert zur Nedden: Do 18 - 22 Uhr, Sa 10 - 14.30 Uhr

Inhaltsverzeichnis

C l u b	
Korrektur & Nachtrag	Seite 2
Fragen & Antworten	Seite 2
Tips	Seite 2
C P / M	
Libraries	Seite 3
T u r b o - P a s c a l	
CP/M und MsDos	Seite 6
Tips zu Dateihandling	Seite 8
R A M x.x	
KLIX-Overlays	Seite 13
R A M 6.0	
Aufruf eines KLIX aus einem Programm	Seite 10
Software-Erweiterungen	Seite 16
RAM 6.0b	Seite 16
Ein paar Anmerkungen	Seite 18
Aliase	Seite 22
Festplatte	Seite 23
L e s e r b r i e f	
Hans Gras, NL-1506	Seite 28
Holger Hansen, 3300	Seite 32
Ulrich Taschke, 7500	Seite 33
Hartmut Traber, 5270	Seite 34
Hartmut Traber, 5270	Seite 35
V A T I C A L	
Eine revolutionäre Programmiersprache?	Seite 36
N e w W o r d	
Grafik und Sonderzeichen	Seite 41

Preis für dieses Info: DM 12.50

(Info wurde zu Dick für DM 12.-; Abonnenten erhalten ja 5% Rabatt, so daß ein Guthaben von DM 12.- genügt).

Redaktionsschluß für's nächste Info: 01.09.90

RAM 6.x (Herbert zur Nedden, 2000)
Auch RAM 6.0a hat noch zwei bekannte Macken: Der Spooler gibt nach einem Timeout einen unmotivierten FormFeed aus und die Uhr mag sich nicht am 8.7.90 stellen lassen. Hier der Patch in Kurzform, der aus RAM 6.0a RAM 6.0b macht (ist auch auf KCLICK:011):

```
GET 100 RAM60.COM
POKE 087D 9F;POKE 2503 90 97;POKE 5ADF 96 97
POKE 9572 C5 CD 23 34 C1 C9;POKE 9578 E5 2A 3E 5B CD 9A 5D E1 C9
SAVE 298 RAM60B.COM S
```

FORMATE.R6 von RAM 6.x (Herbert zur Nedden, 2000)
Folgendes Format ist neu hinzugekommen: 55 = c't SRAM-Disc 512 kB ohne Systemsp. (Man nehme Format 51 und setze die Zahl der Systemspuren auf 0.)

ZCPR 3.3 (Herbert zur Nedden, 2000)
Der ZCPR 3.3 behandelt die Multiple Command Line nicht richtig. Er versaut das Byte, in dem die max. Länge der MCL steht. Damit glauben einige ZCPR 3.3-Utilities wie u.a. ARUNZ (Alisae) an eine zu kurze MCL. (Dieser Mist kam so aus den U.S.A.). Auf KCLICK.011 ist die korrigierte Version.

Da sich gegenüber der letzten Kleinanzeigen und Angebote nur recht wenig geändert hat, hier nur die Neuerungen:

V E R K A U F

Herbert zur Nedden, Sonnenau 2, 2000 Hamburg 76, 040 - 200 87 04
Infos 11 bis 36, gebraucht: DM 75.- frei Haus (VB)

S U C H E

Herbert zur Nedden, Sonnenau 2, 2000 Hamburg 76, 040 - 200 87 04
TP 200

CLUB.910 IN ARBEIT: ca. 111 kB fehlen noch

Diverse ZCPR3-Programme von Hans Gras:

ABORT: Abbrechen einer Kommandozeile. CD35: Change Directory. CRC33: eben deren Berechnung. DIR: 4-spaltige DIR-Anzeige. DUMPDIR3: Dump Directory mit FLUSH-Usernamen und Anzeige doppelt belegter Blöcke. ERROR32: Error Handler. HELP51: Das alte aber schnellere Help. JULDISP: Display Julian date. LDIRZ: Library-Dir. UNSPOOL3: Spooler. SAVEMEM: Speicherbereich auf Disk schreiben.

KLICK.011

Fehlerkorrekturen: RAM 6.0b, FCP, ZCPR 3.3, SIZE, PATH. Interleave-Tabellen aus FORM6. MS 60a (mit Atari/Apricot). HD-Parker. Neuer LOAD61, der das KLIX beim Laden umbenennen kann. Aufruf KLIX aus Assembler-Programmen. Anzeige der DMX 80-Steuerzeichen. CLIPPER: Transport von 80-Zeichen-Karte in Tastaturpuffer. LBREX30P: LBR-Extract, den fehlende CRCs nicht stören. MENU1.62 auf RAM 6 angepaßt mit einigen Ideen dazu. Directory-Editor DIREEDIT. MsgToZ80 (Ablösung von MsgToMac) welches hierarchische Menüs kann und die CP/M-Version von E45, die insbesondere User kennt.

RAM - Patches

Für DM 5.- incl D&P&V kannst Du von mir den Patch RAM 6.0a -> RAM 6.0b und den reparierten ZCPR 3.3 samt dem dazugehörigen ZCBIOS3 erhalten.

HINWEIS in eigener Sache

Ab Anfang 1991 habe ich folgende Anschrift (hoffentlich):

Herbert zur Nedden
Alte Landstraße 21
2071 Siek
Tel.: (04107) ????

Die Kontonummer des MTX User-Club Deutschland bleibt unverändert.

Siek liegt nord-östlich von Hamburg, südlich von Ahrensburg.

Liebe Leserin, lieber Leser,

dieses Mal klappte es mit Beiträgen zum Info wieder mal hervorragend! Vielen Dank den emsigen Autoren.

Eine Bitte zu Artikeln für's Info: Wenn Du Textpassagen einrücken möchtest (z.B. um sie übersichtlich durchzunummerieren), dann verwende dafür bitte entweder ^O^G oder .LM. Mit Leerzeichen einzurücken macht jegliches Überarbeiten zum Greuel! Da ist es immer noch einfacher für mich, Du rückst nicht ein und vermerkst einen entsprechenden Wunsch als Kommentar. Falls Du den Text nicht auf 80 Spalten Breite formatierst, laß bitte die Silbentrennung weg. Ich reformatiere Deinen Artikel nämlich auf 80 Spalten wobei bestehende Silbentrennstellen gelegentlich stören.

Und noch eine weitere Bitte: Wenn Du ein Programm aus dem Club verbesserst oder gar Fehler herausnimmst, wäre es toll, wenn Du den Source so wenig wie möglich ändern könntest - sprich nicht irgendwie reformatierst usw. Just habe ich am FCP eine Kleinigkeit korrigiert, und Hans Gras eine andere. Also beide Versionen zeilenweise verglichen - uff! Hans hatte gründlich reformatiert.

Dieses Info enthält nach langer Pause wieder ein Gesamt-Info-Inhaltsverzeichnis. Peter Würfel meinte dazu:

Die Rubrik 'Software' im Info-Inhaltsverzeichnis ist von mir so aufgebaut, daß (in der Regel) unter Stichwort0 der Programmname und unter Stichwort2 der Ort der Veröffentlichung (also, wenn PD, auf welcher Diskette dieses Programm zu finden ist) eingetragen wurde. Um mir in Zukunft die Arbeit beim Einhacken des Inhaltsverzeichnisses zu erleichtern, bitte ich Euch, immer dann, wenn Ihr etwas über ein bestimmtes Programm schreibt, irgendwo im Text den korrekten Programmnamen (so wie man ihn im Direktory findet) und den Diskettenamen zu erwähnen. Wenn es sich um ein Programm handelt, das es in verschiedenen Versionen gibt, dann nennt bitte auch die Versionsnummer, auf die sich Euer Text bezieht.

Und nochmal der Hinweis: Das Inhaltsverzeichnis kann nur so gut sein, wie Eure Zusammenarbeit! Also: Wenn Ihr etwas sucht und nur mit Mühe findet, weil es unter einem Stichwort eingetragen ist, unter dem Ihr es zuerst nicht vermutet habt, dann laßt es mich doch bitte wissen, damit ich entweder den Eintrag auch unter diesem Stichwort vornehmen bzw. in Zukunft Euren Wunsch berücksichtigen kann.

Euv *Hubert*

Fehler in RAM 6.0

(Herbert zur Nedden, 2000)

RAM 6.0 kann man sicherlich nicht als Buggy (= Wanzling) bezeichnen, da es eigentlich recht fehlerfrei ist - und das bei über 500 kiloBytes Assembler-Source (die Kommentare mitgerechnet). Leider hat die Vergangenheit gezeigt, daß doch noch kleine aber auch harmlose Fehler in RAM 6.0 stecken, deren Entdeckung eher per Zufall erfolgt. Z.B. DATE hatten Olaf und ich getestet - nur halt nicht am 8.7.90 - und so übersahen wir den Fehler (der in diesem Info seine Beseitigung erfährt). Es gibt allerdings auch Fehler wie z.B. den Zweistrich-Cursor, von dem ich zwar weiß, den aber nicht reproduzieren kann. Daher möchte ich die Fehlerfindigen unter Euch bitten, mir möglichst eine Diskette zu schicken, von der ich booten und den Fehler nachvollziehen kann.

C l u b: Korrektur & Nachtrag / Fragen & Antworten / Tips**Korrektur & Nachtrag**Installation des Dia-Programms (CLUB.049): (Peter Würfel, 7262)

In der Dokumentation fehlt der Hinweis, daß in DIA.CMD nicht nur die Programmzeile 7 (Kommentarzeilen zähle ich nicht als Programmzeilen), sondern auch die Programmzeile 35 entsprechend der eigenen Gerätekonfiguration geändert werden muß. In Zeile 35 steht:

```
STOR 'A' TO mp
```

Dort den Laufwerkbuchstaben eintragen, auf dem sich die Programme befinden. (Übrigens, ich würde mich freuen, mal 'ne Reaktion auf dieses Programm zu bekommen).

RAM 6.0, Flow Command Package (FCP.SYS): (Herbert zur Nedden, 2000)

Zwei kleine Fehler sind drin. SHOW zeigt Mist an (behooben durch Hans Gras) und der FCP murkst mit dem Pfad rum. Letzteres merkst Du, wenn Du nach dem Booten den Pfad änderst und mit IF arbeitest. Lösung auf KCLICK.011.

Fragen & Antworten

A: (Ulrich Taschke, 7500)

Antwort zur Frage von Anton Reiser in Info 37-1

Du hast Glück! Der 8751-8 ist der Vorgänger des 8751H und hat noch kein "Security-Bit". Wenn das nämlich programmiert ist, dann kann man das EPROM auf dem Prozessor nicht mehr auslesen. Ein idealer Softwareschutz!

Der 8751 gehört zur Familie des 8051, auf dem ich früher einige Entwicklungen gemacht habe. Er ist sehr universell und relativ einfach zu verstehen.

Ich schicke Dir über Herbert ein paar Kopien, mit denen Du Dir einen guten Überblick verschaffen kannst.

Ich hatte übrigens mal vor, für den ROM-losen Prozessor (8031) auf MTX-Basis eine Art Mini-Entwicklungssystem zu machen. Genauer: 8051-Crossassembler mit M80-Macros; in Verbindung mit dem c't-PEPS einen Mini-Monitor usw. Aber wie so vieles habe ich das aus Zeitgründen wieder fallengelassen. Ich schätze jedoch, daß Du um ähnliches nicht herumkommen wirst, wenn Du wirklich das Programm auf Deinem Centronics-Interface auslesen und ändern willst. Falls Du es machst, hätte ich schon mal Interesse....

F: (Andreas Fischer, CH-4303)

Wie kann das Autostartsystem erweitert werden, damit nach einem Absturz nach SIDISC und CONFIG B:7,F:51 auch noch COLDBOOT F: und XYZ.COM ausgeführt werden? Ich bewecke damit eine automatische Wiederaufnahme des laufenden Programmes bei einem Absturz. Da mein Wettercomputer immer laufen soll, wäre eine automatische Wiederarufnahme der Arbeit vorteilhaft.

Tips: Tastatur

(Herbert zur Nedden, 2000)

Sollte bei Dir eine Taste nicht mehr so richtig wollen, dann muß gar nicht mal die Taste selbst hinüber sein. Eher hat deren Lötkontakt Schuld. Das andauernde Herumgehämmere ist sicherlich nicht gerade der Lebensdauer der Tastatur dienlich.

Wenn Du Glück hast, dann hast Du die Taste lediglich aus deren Lötstelle herausgeklopft - dann genügt einfaches Nachlöten der Kontakte in der Tastatur. Solltest Du Pech haben (wie ich), dann hat sich die Lötstelle als sehr stabil erwiesen - die Leiterbahn jedoch nicht. Dann hat sich nämlich die Lötstelle samt dem zugehörigen Lötauge von der Leiterbahn getrennt, was kaum zu sehen ist. Daher solltest Du darauf beim Nachlöten achten, und ggf. einen kleinen Draht zur Wiederherstellung der Verbindung spendieren.

Hilft all dies nicht, dann ist es vielleicht doch an der Zeit, mal die Taste selbst zu ersetzen. Dazu wird der Lötzinn auf der Unterseite soweit entfernt, daß die Pins der Taste lose sind. Nun wird oben die Tastenkappe dieser Taste und seiner Nachbarn entfernt und die Taste selbst herausgeholt.

C P / M: Libraries**Libraries**

(Herbert zur Nedden, 2000)

Hartmut Traber meinte unlängst, daß er Libraries und die dadrin auch noch komprimierten Dateien auf Public-Domain-Disketten nicht befürwortet. Das mit dem Ansehen und Auseinandernehmen sei doch recht mühselig - warum denn nicht einfach weniger auf die Disketten?

Irgendwo kann ich das schon verstehen! Aber ...

Eine Library ist eine Datei, in der mehrere einzelne Dateien drin stecken. Sie stellen also eine Art von Unterdirektory (= Subdirektory) dar. Der Dateityp ist .LBR. Sie bieten folgende Vorteile gegenüber der Speicherung einzelner Dateien auf Diskette:

Übersichtlichkeit: Ich kann zusammengehörige Dateien in eine Library stecken - im normalen Disketten-Direktory taucht dann nur der Name der Library auf.

Disketten-Platz: Auf Diskette wird der Platz für Dateien in Blöcken belegt. Eine Datei, die 10 Bytes enthält, belegt auf Diskette (bei unseren üblichen Formaten) 2kB. In Libraries hingegen wird der Platz in Sektoren vergeben, so daß die 10-Byte-Datei nur 128 Bytes in der Library belegt. Gerade bei vielen kleinen Dateien macht sich diese Einsparung sehr bemerkbar.

Direktory-Platz: Libraries enthalten ihr eigenes Direktory in sich drin. Beim Anlegen einer Library kann ich festlegen, wieviele Einträge ihr Direktory haben soll - bei Disketten ist die Anzahl der Direktory-Einträge schon vorgegeben. Soll eine Library 400 Dateien aufnehmen, so hapert es nicht am Direktory-Platz.

Bekanntlich gibt es nichts gratis! Libraries werden leider nicht direkt vom BDOS unterstützt - sozusagen als logische Laufwerke, Unterverzeichnisse o.ä. Auf Libraries kannst Du daher nur unter Zuhilfenahme von Programmen zugreifen. Will ich eine Datei aus einer Library bearbeiten, muß ich sie i.a. zuerst rausholen.

Hier eine Auflistung von einigen Programmen mit deren mehr-oder-weniger typischen Namen, mit denen Libraries bearbeitet werden können. Erkennen kannst Du diese Utilities meist daran, daß deren Name mit einem L = Library beginnt. Gelegentlich beginnt der Programmname auch mal mit LBR statt nur mit einem L.

- LDIR Ein Programm, welches das Library-Direktory anzeigt. Daher der Name. Es gibt allerdings mittlerweile auch Programme vom Typ Super Direktory (mit Namen SD), die das auch können.
- LTYPE TYPE-Befehl für eine Datei in einer Library.
- LGET Herausholen (GET) von Dateien aus einer Library.
- LBREXT Ebenfalls Herausholen (EXTRACT) von Dateien aus einer Library.
- LPUT Hineinstopfen (PUT) von Dateien in eine Library.
- LRUN Ausführen (RUN) eines Programms, welches in einer Library steht. LRUN holt das Programm aus der Library und startet es. Üblicherweise ist ja der CCP (bei uns der ZCPR) dafür zuständig das Programm zu laden und aufzurufen.
- LU Library-Utility. Mit LU kannst Du i.a. die Aufgaben von LDIR, LTYPE, LGET und LPUT und einiges anderes mit Libraries anstellen.

C P / M: LibrariesJetzt wird es etwas technischer:

Auf Disketten wird der Platz für Dateien wie gesagt in Blöcken belegt. Dabei können die einzelnen Böcke - äh Blöcke - lose über die Diskette verteilt sein. Das hat den Vorteil, daß Löcher in der Blockbelegung nicht weiter stören.

In Libraries ist das anders. Die Sektoren einer Datei müssen innerhalb der Library hintereinander stehen. Lösche ich eine Datei aus einer Library entsteht ein Loch, welches nur dann wieder belegt wird, wenn eine neue Datei in die Library hineinkommt, die in das Loch reinpaßt. Das hat zur Konsequenz, daß eine Library gelegentlich (nach diversen Löschungen und Einfügungen) reorganisiert werden muß. Library-Utilities nennen das 'pack'. Das funktioniert dadurch, daß automatisch eine neue leere Library angelegt und die Dateien aus der einen in diese neue kopiert werden. Anschließend wird die alte gelöscht und die neue umbenannt. Das ist jedoch nicht mal so tragisch. Normalerweise sind die Dateien, mit denen ich arbeite und die sich dabei ändern oder auch mal verschwinden, nicht in Libraries.

Andererseits lassen sich die Dateien in Libraries meist schneller lesen als von Disketten, da sie am Stück drin stecken.

Wenn der ZCPR ein Kommando erhält, mit dem dieser nichts anzufangen weiß, d.h. wenn dieser das zugehörige Programm nicht findet, so startet er einen ECP (Extended Command Processor): CMDRUN. Verwende ich LRUNZ (= LRUN, ZCPR-Version) als CMDRUN sucht LRUNZ das gesuchte Programm in einer Library (COMMAND.LBR) und startet es, falls gefunden. Damit ist die Library COMMAND.LBR für mich wie ein Laufwerk, von dem ich allerdings nur .COMs ausführen kann. Für andere Zugriffe brauche ich auch weiterhin Programme (LDIR, LPUT usw.). Üblicherweise ist unter RAM 6.x CMDRUN sicherlich ARUNZ (Alias-Laufenlassprogramm). Dieses verzweigt dann bei Mißerfolg nach XRUN (Extended Run) und dieses ggf. nach LRUNZ bzw. LX (= Library Execute).

Jan Brederke ist übrigens dabei, ein KLIX-Overlay zu schreiben, mit dem eine Library als Laufwerk konfiguriert werden kann. Dieses Laufwerk kann wie ein Disketten-Laufwerk über die normalen BDOS-Funktionen angesprochen werden, allerdings nur lesend. Damit ist LRUN hinfällig und ich kann in meiner COMMAND.LBR-Library auch andere Dateien als .COMs drin haben. (Prototyp läuft.)

Libraries haben sich im Bereich der Public-Domain Software mehr und mehr durchgesetzt. Zusammengehörige Dateien werden einfach in eine Library gesteckt und schon ist die Verteilung leichter und die Chose ist übersichtlicher und kostet weniger Platz. Wenn wir Programme in Libraries bekommen, geben wir sie auch so weiter.

Obendrein werden die Dateien in Libraries meist noch gecruncht - ebenfalls aus Platzgründen. Da z.B. LBREXT29 die Dateien beim Herausholen gleich dekomprimieren kann und einige Library-Utilities, mit denen man erst mal in die Library hineinsieht, bei der Anzeige ebenfalls dekomprimieren, stört das nicht allzusehr.

Übrigens haben wir eine Großteil der aus den U.S.A. erhaltenen Software in eben diesen Libraries in gecrunchter Form erhalten - und folglich auch so weitergegeben. In einigen der Libraries war sogar vermerkt, daß die Library (oder zumindest deren Inhalt) nur komplett weitergegeben werden darf.

Bei RAM 6.0 habe ich allerdings die Dateien in den Libraries ausnahmsweise gelZH-cruncht, um alles, was mit sollte auf drei Disketten unterzubringen. Dummerweise unterstützt nur LBREXT29 diese Kompressionsart!

C P / M: LibrariesNun zur Bedienung einiger Library-Bearbeitungs-Programme:

Also erst mal das Hineinstopfen von Dateien in eine Library. Die Dateien sind DATEI1, DATEI2.ZZZ (also gecruncht) und DATEI3.YYY (geLHZcruncht); sie sollen in die Library LIBRARY.LBR. Hier der Ablauf:

G1>lput library *.*

Alles (.*) vom akt. Directory in die Library LIBRARY.LBR hineinstecken. (Mehr als die drei o.g. Dateien sind zufällig nicht im akt. Directory)*

LPUT, Version 1.6p

Making library: G1:LIBRARY.LBR,

Putting file G1:DATEI1. (07/19/90) in library,

Putting file G1:DATEI2.ZZZ (07/19/90) in library,

Putting file G1:DATEI3.YYY (07/19/90) in library,

G1:LIBRARY.LBR closed, operation complete.

Das Herausholen mit gleichzeitiger Dekompression geht wie folgt:

G1>lbrect library 2:*. * u

Alles (.*) aus der Library LIBRARY.LBR herausholen, dabei dekomprimieren und nach User 2 stellen.*

File: DATEI1. (07/19/90) ---> G2:

File: DATEI2.ZZZ (07/19/90) ---> G2:DATEI2.

File: DATEI3.YYY (07/19/90) ---> G2:DATEI3.

Mit QL kannst Du bequem in Libraries hineinsehen. Die Bedienung ist interaktiv. Einfach die Zahl neben der angezeigten Datei eingeben und schon wird sie angezeigt oder im Falle einer Library eröffnet und die darin enthaltenen Dateien angeboten. Allerdings kann QL die LZH-komprimierten Dateien nicht automatisch bei der Anzeige dekomprimieren.

LBREXT29 hat leider eine kleine Tücke! Es erwartet, daß die CRC-Prüfsummen der Dateien in der Library stimmen. Dummerweise denken die älteren Library-Programme nicht daran, diese CRC-Prüfsummen einzutragen. Stimmt die Prüfsumme nicht (was auch der Fall ist, wenn sie fehlt), so ist LBREXT29 der Meinung, da sei etwas Faulm im Staate und löscht die entladene Datei wieder.

In diesem Fall bleibt nur noch LU oder NULU. Damit entlädst Du die Library komplett. Dann löschst Du die alte Library und erstellst sie mit LPUT neu. Oder:

Es gibt nun LBREX30P (auf KCLICK.011), welches gegenüber dem LBREXT29 folgende Unterschiede hat:

- Es unterstützt P2DOS-Zeiteinträge (daher das 'P' am Ende des Namens)
- Es ignoriert den CRC, so er fehlt! Damit können nun auch alte Libraries mit LBREX30P auseinandergenommen werden.
- Die Dateien werden standardmäßig dekomprimiert. Mit U wird dies nun verhindert. Immer an das 'U' zu denken war mir zu dumm!
Wenn Dich das stört: In LBREX30P.COM steht ab Adresse 112h UCPRS>. Direkt dahinter (Adresse 188h) steht Offh. Setzt Du hier 00h rein, so ist das mit dem 'U' wie früher.

Der Aufruf von LBREX30P ist mit dem obigen von LBREXT bis auf das 'U' identisch. Sinnvollerweise solltest Du eh LBREX30P.COM in LBREXT.COM umbenennen. **ACHTUNG:** Hast Du LBREXT in irgendwelchen Aliassen muß Du ein evtl. enthaltenes 'U' für entkomprimieren entfernen, da das im ausgelieferten LBREX30P default ist.

T u r b o - P a s c a l : CP/M und MsDos

'Portable' Programmierung

(Herbert zur Nedden, 2000)

Gelegentlich will man ja vielleicht ein Programm für einen DoMessDos-Rechner schreiben. (Natürlich nur, um es dann zu verkaufen!) Das alleine ist jedoch eigentlich kein Grund, sich vor so eine Kiste zu setzen, oder ?

Vor längerer Zeit habe ich genau so etwas tun wollen, wobei das Programm sowohl auf meinem MTX (für mich) als auch auf einem MsDos (zum Verkauf) laufen sollte. Schreibt man das Teil richtig sauber, d.h. nur unter Verwendung von in Turbo-Pascal eingebauten Functions und Procedures für die Ein- und Ausgabe ist das ein Leichtes - wenn da nicht die Bildschirm-Attribute und Sonderzeichen wären.

Die Lösung boten für mich folgende zwei Include-Dateien:

CP/M.INC

```
type Farbe = (NormColor,BrightColor,DarkColor,TitleColor,CopyriColor);
```

```
procedure TextColor(c: Farbe);
begin
  write(chr(6)+chr(0));
  case c of
    TitleColor : write(^U);
    NormColor   : write(^R);
    DarkColor   : write(^R);
    BrightColor : write(^T);
    CopyriColor : write(chr(6)+chr(16));
  end;
end;
```

```
Procedure Pfeile;
begin
  write(#27,'GJK',#27,'S');
end;
```

MsDos.INC

```
const NormColor   = White;
      BrightColor  = LightGreen;
      DarkColor    = Green;
      TitleColor   = LightRed;
      CopyriColor  = Yellow;
```

```
Procedure Pfeile;
begin
  write(chr(24),chr(25));
end;
```

TextColor ist die für die Einstellung der Bildschirm-Farbe in der MsDos-Version von Turbo-Pascal enthaltene Procedure; die Farben sind hierfür auch gleich noch als Konstanten vordefiniert.

T u r b o - P a s c a l : CP/M und MsDos

Ich verwende in meinem Programm die Procedure TextColor für die Umschaltung der Farbe bzw. Helligkeit der Bildschirm-Anzeige. Und damit ich nicht allzu festgelegt bin, habe ich mir folgende Farben-Namen ausgedacht:

NormColor, BrightColor, DarkColor, TitleColor, CopyriColor

Will ich helle Schrift, so codiere ich:

```
TextColor(BrightColor);
```

Befehle wie writeln('^T'Hallo'^R);

sind tabu. Statt dessen sollte es so aussehen:

```
TextColor(BrightColor); writeln('Hallo'); TextColor(NormColor);
```

TitleColor und CopyriColor habe ich für die nämlichen Textpassagen verwendet, damit ich mich jederzeit leicht umentscheiden konnte.

Pfeile hingegen gibt schlicht und ergreifend zwei Pfeile auf dem Bildschirm aus, da deren Ausgabe auf dem MTX anders als unter MsDos erfolgen muß. Bei uns via Grafik-Zeichensatz - beim MsDos hingegen sind das 'normale' Zeichen.

Die beiden o.g. Include-Dateien sind sicherlich nicht der Weisheit letzter Schluß! Dafür geben sie Dir einen Anhalt, wie Du zu Werke gehen kannst.

Willst Du gar mit Windows (= Fenstern) 'portabel' programmieren, so müßtest Du die Window-Routine der MsDos-Version (sie heißt Window) auf der Basis unserer Window-Routine ebenso nachbilden. Das macht allerdings nur unter RAM 4.x oder besser Sinn, da erst ab dieser Version Windows vom Bildschirmtreiber unterstützt werden. Dann ist es jedoch fast banal, da die MsDos-Window-Routine lediglich ein Fenster definiert. Von Retten und Schließen ist nicht die Rede (zumindest nicht bei Turbo-Pascal 3.0)!

Eh ich's vergesse. Das Programm soll ja sowohl auf dem MTX als auch auf dem MsDos kompiliert werden. Da es keinen Spaß macht, mit (. und .) an Stelle der [und] und ohne Umlaute zu leben, macht es Sinn, beim Wechsel MTX <-> MsDos einen Zeichensatz-Konverter auf den Source loszulassen, der die Umlaute umsetzt, da unsere Ä und Ü unter Turbo-Pascal mit [] identisch sind, MsDos jedoch für Ä und Ü andere Ascii-Codes als für [und] hat. Die Lösung dieser Aufgabe übernimmt mein Progrämmchen CONVERT (u.a. CLUB.052), welches folgendes tut:

Konvertierung CP/M --> MsDos

```
Ä Ü ----> Ä Ü, falls in einem String enthalten
           [ ], sonst
```

Konvertierung MsDos --> CP/M

```
Ä Ü ----> Ä Ü
[ ] ----> Ä Ü
```

D.h. In Strings stehende Ä Ü [] gelten stets als Ä Ü. Damit konnte ich gut leben, da ich keinen Bedarf hatte, [] anzuzeigen.

T u r b o - P a s c a l : Tips zu Dateihandling

Tips zu Dateihandling

(Herbert zur Nedden, 2000)

In einem Programm wollte ich zwei Dateien zum einen schnell, indiziert aber auch Diskettenplatzsparend verarbeiten. Die Daten hatten folgendes Format:

1. Eine Datei mit je drei Integer-Zahlen pro Satz und
2. Eine Datei mit Strings variabler Länge von 3 bis 255.

Folgende Definitionen der Dateien sind naheliegend:

```
Type TSatz1: record
    Zahl1: integer;
    Zahl2: integer;
    Zahl3: integer;
end;
Type TSatz2: string[255];

Var Datei1: File of TSatz1;
    Datei2: File of TSatz2;
```

Für Datei1 ist das auch O.K., Datei2 hingegen wird so unangenehm groß. Daher habe ich Datei2 wie folgt definiert:

```
Var Datei2: Text;
```

Also als Datei, die mit ReadLn eingelesen werden kann. Da Datei2 durchaus 500 Zeilen enthält kann ich sie nicht in einen Puffer vom Typ Array of String[255] einlesen, da ich dafür zu wenig Platz habe. Jedesmal nachlesen geht aber auch nicht, da ich im Programm laufend hin und her blättern muß.

Ich habe mich daher zu folgendem Vorgehen entschlossen:

1. Beide Dateien werden mit BlockRead auf einen Schlag eingelesen, was bei meinen Daten geht, da ich weiß, daß jeweils 10000 Bytes (ja Zehn-Tausend!) genügen. Dafür muß allerdings ich beide Dateien als File definieren.
2. Die Verarbeitung der Datei1 ist leicht, da sie ein Array vom Typ Satz1 darstellt. Allerdings ist zu beachten, daß in der Datei selbst vor den Daten zwei Integer Zahlen (Länge der Datei und Länge eines Satzes) stehen.
3. Datei2 wird in ein Array of Char eingelesen und 'zu Fuß' durchgearbeitet.

Hier ein Auszug aus meinem Programm:

```
Type TSatz1: record
    Zahl1: integer;
    Zahl2: integer;
    Zahl3: integer;
end;
Type TSatz2: string[255];

Var Datei1: File;
    Datei2: File;

Var Satz1: TSatz1;
    Satz2: TSatz2;

Var Daten2: array[0..10000] of Byte;
    Daten1: array[0..500] of TSatz1;
    Header1: array[1..2] of integer;
```

T u r b o - P a s c a l: Tips zu Dateihandling

```

Procedure GetFiles;
  { ACHTUNG: Diese Procedure prueft nicht, ob die Datei in den Puffer passt }
var i:      integer;

begin
  { Lies Datei1 an die Adresse Header1. Damit landen die beiden Integers,
    die vorne in dieser Datei stehen, da sie als File of TSatz1 erstellt
    wurde in Header1 und die eigentlichen Daten in Daten1. Beachte, dass
    Turbo-Pascal die Datenbereiche von oben nach unten im Speicher anlegt! }
  assign(Datei1,'DATEI1.NUM');
  reset(Datei1);
  blockread(Datei1,Header1,FileSize(Datei1),i);

  { Lies Datei2 direkt in den Puffer, da es sich dabei um eine normale Text-
    Datei handelt, die daher die beiden Integers vorne drin nicht hat. }
  assign(Datei2,'DATEI2.TXT');
  reset(Datei2);
  blockread(Datei2,Daten2,FileSize(Datei2),i);
end;

Function TextZeile(Nummer: integer): TSatz2;
{ Hole Zeile Nummer aus dem Textpuffer Daten2. Dabei ist die erste Zeile bei
  mir Zeile Nummer 0.
  ACHTUNG: Diese Function hat KEINE Ende-Abfrage, da ich sie nicht brauchte! }

var i,t: integer;
    s:  TSatz2;

begin
  if Nummer < 0 then
    TextZeile := ''
  else
    begin
      t := 0;           { Textpointer = 0 }
      i := 0;          { Zeilenzaehler = 0 }
      while (i <> Nummer) do { Wandere, bis Anfang Zeile Nummer gefunden }
        begin
          repeat
            t := succ(t);
          until (Daten2[t] = ^J); { Zeilenende = CR LF, also LF = ^J }
          t := succ(t);          { Ueberspringe ^J }
          i := succ(i);          { Zeilenzaehler +1 }
        end;
      s := '';
      while (Daten2[t] <> ^M) do { Hole Zeile bis einschliessl. CR = ^M }
        begin;
          s := s + Daten2[t];
          t := succ(t);
        end;
      TextZeile := s;
    end;
end;

```

Ggf. kann es Sinn machen, nach dem Einlesen der Dateien deren Länge zu ermitteln, damit nicht aus Versehen über das Ende hinaus in den Puffern herumgearbeitet wird. Bei der ersten Datei steht deren Länge in 'Anzahl der enthaltenen Records' in Header[1]. Bei Datei2 mußt Du den Puffer solange durchflöhen, bis Du das Ende-Zeichen ^Z (= 1Ah) findest oder das Dateiende erreichst. Die Länge der Datei in Bytes ist 128*FileSize(Datei2).

R A M 6.0: Aufruf eines KLIX aus einem Programm**Aufruf eines KLIX aus einem Programm**

(Herbert zur Nedden, 2000)

Wozu? Ein KLIX kann ich leicht via <SHIFT-ESC> gefolgt von der entsprechenden F-Taste aufrufen. Nun ja:

Für meinen Directory-Editor (dessen Fertigstellung zur Zeit an der Dokumentation und Olaf's Test hängt) wollte ich ein Help einbauen. Üblicherweise wird an diversen Stellen im Programm eine bestimmte Taste abgefragt. Wird sie gedrückt, so wird die Hilfe angezeigt und anschließend der ursprüngliche Bildschirm mühsam wiederhergestellt.

Ich wollte eine Lösung, die folgende Spezifikationen erfüllt:

1. Keinen Platz im .COM verschwenden - schließlich kenne ich mein Programm und brauche das Help nicht.
2. Das Help soll sich jeder leicht ändern können. Da wäre z.B. eine Übersetzung in andere Sprachen denkbar.
3. Der Aufruf soll möglichst leicht ins Programm eingebunden werden können.

Hier meine Lösung: (Ist auf KCLICK.011 als Source)

; Demo-Programm:

```

Loop:          call   HKbdIn
               cp     27
               ret    z
               jr     Loop

               INCLUDE CALLKLIX

ProgName:     db     'DirEdith'
SaveArea     equ   $

```

CALLKLIX.MAC

```

;*****
;*
;* Aufruf eines KLIX-Overlays aus einem Programm heraus *
;*
;* Herbert zur Nedden, 25.07.90 *
;*
;* Ablauf: Suchen des KLIX in der KLIX-Tabelle *
;* Falls nicht gefunden: Ade *
;* Retten des Bildschirms *
;* Aufruf des KLIX *
;* Restaurieren des Bildschirms *
;*
;* Einschränkungen: Das KLIX darf keine BDOS-Calls ab- *
;* setzen, da die entsprechenden Vari- *
;* ablen nicht gerettet werden. *
;* Diese Routine geht davon aus, daß *
;* der Platz zum Retten des Bildschirms*
;* an der übergebenen Adresse verfügbar*
;* ist. Es wird der gesamte Bildschirm *
;* gerettet. *
;*
;*****

```

R A M 6.0: Aufruf eines KLIX aus einem Programm

```

; RAM-Einsprünge
BnkCall      equ    0f90ch          ; Bank-Call
SavScr       equ    0f924h         ; Rette Bildschirm
ResScr       equ    0f927h         ; Restauriere Bildschirm
FKbdIn       equ    0f93fh         ; Tastatur mit F-Tasten
Utility      equ    0f95dh         ; Utility
; Utility-Funktionen
ScK1Nm       equ    65             ; Suche KLIX-Namen
; RAM-Variablen
VarPtr       equ    0ffc6h         ; Pointer auf RAM-Variablen
KliFlg       equ    24h           ; KLICK-Flag
; Bildschirm-Variablen
Spalts       equ    0f42fh         ; Breite des Bildschirms
Zeilen       equ    0f430h         ; Höhe des Bildschirms
; Puffer im COMMON
Puffer       equ    0f980h         ; Puffer im COMMON

```

```

; Hier meine Tastaturabfrage HKbdIn = Help-Keyboard-Input
; -----

```

```

HKbdIn:      call    FKbdIn          ; RAM-Tastaturabfrage
              cp      '?'           ; Help ?
              ret     nz            ; nö
              ld     hl,SaveArea     ; HL/B: Platz zum Retten
              ld     b,80h           ;       des Bildschirms
              ld     de,ProgName     ; DE: name des KLIX
              call   CallKlix
              jr     HKbdIn          ; Roger, over

```

```

; Suche KLIX und rufe es auf, wenn gefunden
; -----

```

```

CallKlix:    ; IN: HL/B: Platz, wohin Bildschirm gerettet werden soll
              ; DE: Zeiger auf zu suchenden KLIX-Namen
              ; OUT: Carry: KLIX wurde gefunden und aufgerufen
              ; NoCarry: KLIX wurde nicht gefunden

              ; 1. Merke Platz zum Retten der Glotze
              ld     (ScrArea),hl
              ld     a,b
              ld     (ScrBank),a

              ; 2. Verschieben des gewünschten Programmnamens in den COMMON
              ; (da die Utility-Funktionen von RAM 6.0 aus Versehen deren
              ; Daten auf der aktuellen Bank, also Bank 1 erwarten.)
              ex     de,hl
              ld     de,Puffer
              push   de
              ld     bc,8
              ldir
              pop    de

              ; 3. Suchen des Namens in der KLIX-Tabelle
              ld     c,ScK1Nm
              call   Utility          ; DE ist noch richtig

```

R A M 6.0: Aufruf eines KLIX aus einem Programm

```

; 4. Falls nicht gefunden: Ade
ret    nc

; 5. Eintragen der Zieladresse in den Bank-Call
ld     (Adresse),de
ld     (Bank),a

; 6. Retten des Bildschirms
call   SetScrRegs
call   SavScr

; 7. Aufruf des KLIX
; (KLICK-Flag so setzen, daß KLICK als aktiv gilt, damit
; <SHIFT-ESC> nicht die Interrupts beschleunigt - immerhin
; wird ein KLIX-Overlay aufgerufen, und das ohne, daß RAM 6
; das offiziell erfährt.
; Für den Bank-Call muß der Stack in den COMMON!)
ld     iy,(VarPtr)
ld     a,(iy+KliFlg)
push   af
ld     (iy+KliFlg),0ffh      ; Simuliere KLICK aktiv
ld     (SpSave),sp
ld     sp,Puffer+70h
call   BnkCal
Adresse: dw    0
Bank:    db    0
         ld    sp,0
SpSave   equ   $-2
         ld    iy,(VarPtr)
pop      af
ld     (iy+KliFlg),a      ; KLICK-Flag restaurieren

; 8. Restaurieren des Bildschirms
call   SetScrRegs
call   ResScr
scf    ; Carry = O.K.
ret

ScrnArea: dw    0
ScrnBank: db    0

SetScrRegs: ; Diese Routine setzt die Register für SavScr und ResScr:
ld     de,(ScrnArea)      ; Platz für Bildschirmvars.
ld     hl,50              ; Länge Parameter
add    hl,de
push   hl
pop    iy                 ; Dahinter Bildschirminhalt
ld     ix,0               ; Ab Ecke oben links
ld     a,(Spalts)
ld     b,a                ; Breite
ld     a,(Zeilen)
ld     c,a                ; Höhe
ld     a,(ScrnBank)      ; Bank
ret

```

Ach so: Das Help ist natürlich ein KLIX-Overlay mit dem Namen DirEdith.

R A M x.x: KLIX-Overlays**Übersicht über die KLIX-Overlays**

(Herbert zur Nedden, 2000)

Es gibt mittlerweile eine Flut von KLIX-Overlays. Daher habe ich mich mal entschlossen, eine Übersicht über diese Teile zusammenzustellen und gleich zu vermerken, wo die neueste Version zu finden ist.

Name: so heißt das Teil
 Typ: je nachdem, ob es als .COM seinen Loader enthält, oder als .KLX auf der Diskette steht und mit Olaf's LOADER geladen wird.
 Wo: 'nnn': Nummer der KLIICK-PD, auf der die neueste Version drauf ist bzw. 'n.n': RAM-Version, mit der die neueste Version ausgeliefert wurde.
 RAM: Mindest-RAM-Version, unter der das Teil läuft.
 Ist diese Angabe *kursiv*, so ist das KLIX nicht mehr aktuell.
 Autor: der, von dem das Teil ist
 Funktion: das nämliche.

Name	Typ	Wo	RAM	Autor/Funktion
ASCII6	.KLX	009	4.1	Olaf Krumnow, Uwe Grass, Herbert zur Nedden Anzeige der Hex- und Dezimalcodes aller Zeichen von 00h - 0ffh. Darstellbare Zeichen werden in allen drei Zeichensätzen (Standard, Alternativ und Grafik) angezeigt, von Ctrl-Codes wird deren Wirkung bei der Ausgabe auf den Bildschirm unter RAM 6.x angegeben.
CALC	.KLX	004	4.1	Bernd Preusing, Herbert zur Nedden Taschenrechner mit Grundrechenarten sowie Quadrat und Wurzel. Die Zahlen werden wahlweise binär, oktal, dezimal oder hexadezimal dargestellt. Rechengenauigkeit ist sehr hoch. Taschenrechner-Fensterchen kann über die CP/M-Bildschirmanzeige gelegt werden.
CFIG5	.KLX	005	4.?	Michael Keßler Config mit Unterstützung des Format-ID's. WICHTIG: Läuft nicht unter RAM 6.x! Seine Funktion ist jedoch in RAM 6.x fest integriert.
CLIPPER	.COM	011	4.1	Olaf Krumnow Ausschnitte der 80Zeichen-Karte in das ClipBord stellen, welches als Funktionstaste Shift-AlphaLock fungiert, d.h. der Inhalt des ClipBords wird bei Drücken von Shift-AlphaLock wie eine andere Funktionstaste ausgegeben.
CRT40	.KLX	004	4.?	Hans Gras Treiber für den VS 4. Ob er läuft und was er alles kann, ist mir nicht genau bekannt.
DBTAST	.KLX	007	4.1	Peter Würfel Anzeige dBASE-Control-Befehle und den zugehörigen F-Tasten bei Peter's Belegung.
DC	.KLX	007	4.5	Olaf Krumnow Directory-Cache, d.h. das Overlay speichert das Directory von Disketten-Laufwerken im RAM zwischen, was Lesezugriffe erheblich beschleunigt. WICHTIG: Läuft nicht unter RAM 6.x! Seine Funktion ist jedoch in RAM 6.x fest integriert.
DIJEY	.KLX	008	4.5	Olaf Krumnow Disc-Jockey, ein Nsweep-Ersatz im KLIICK. DiJey kann u.a. Dateien kopieren (mit Verify), anzeigen, drucken, umbenennen, löschen, entsqueezen und die Dateiattribute verändern. Dabei können diese Funktionen leicht für einzelne oder mehrere Dateien ausgeführt werden.

R A M x.x: KLIX-Overlays

Name	Typ	Wo	RAM	Autor/Funktion
DIREDITH	.KLX	???	4.1	Herbert zur Nedden Help zu DirEdit, dem Directory-Editor. DirEdit läuft nur unter RAM 6.x oder besser. (Noch nicht fertig!)
DJHELP	.KLX	008	4.5	Olaf Krumnow Help zu DiJey. Dieses KLIX kann auch direkt von DiJey aus aufgerufen werden.
DMX80	.KLX	011	4.1	Jan Brederke Anzeige der Befehle, die der Drucker DMX 80 versteht.
DOCTOMAC	.KLX	---	4.1	Herbert zur Nedden (CLUB.040). Ist durch MsgToZ80.KLX ersetzt.
E45	.KLX	009	4.5	Herbert zur Nedden MTX-Edit, d.h. mein Non-Document-Editor, der speziell für den Memotech ausgelegt wurde, als KLIX-Overlay. Dabei kann der Editor selbstverständlich direkt auf die Disketten oder auch anderen Laufwerke zugreifen, d.h. Dateien einlesen/sichern. P2DOS-Errors werden abgefangen, lösen also keinen Warm-Boot aus (wenn bei dem KLIICK-Blinkfenster, welches Diskettenzugriffsfehler anzeigt A=Akzeptieren gedrückt wird, so es erscheint).
EDICTHLP	.KLX	005	4.1	Herbert zur Nedden Aufrufe der Edicta-Grafik-Software mit Parametern.
EHELP45	.KLX	009	4.1	Herbert zur Nedden Help zu E45.KLX, damit der Hilfe-Bildschirm nicht unnötig Platz im KLIICK-Heap verbrät.
FESTC	.KLX	007	4.5	Claudio Romanazzi Festplatten-Treiber für OMTI-Controller. WICHTIG: Läuft nicht unter RAM 6.x! Seine Funktion ist jedoch in RAM 6.x fest integriert.
FORM5	.KLX	007	4.3	Michael Keßler Disketten-Formatierprogramm mit Unterstützung des Format-ID's. WICHTIG: Läuft nicht unter RAM 6.x! Ist durch Formstar (FS?DX.KLX und MKF?DX.KLX) ersetzt.
FSFDX	.KLX	6.0	6.0	Michael Keßler, (Herbert zur Nedden) FDX-Version von FormStar: Disketten-Formatierprogramm, welches ca. 100 Formate formatieren kann. Unterstützt das Format-ID.
FSFDX	.KLX	6.0	6.0	Michael Keßler, (Herbert zur Nedden) dto. für die SDX
GDC60	.KLX	010	6.0	Claudio Romanazzi Treiber, der es ermöglicht, eine Edicta-Grafikkarte als RAM-Floppy zu betreiben.
HARD301	.COM	008	4.1	Olaf Krumnow Hardcopy für 80Zeichen (mit/ohne Berücksichtigung der Attribute), VS4 (auch in doppelter Größe und für einige Spezialfälle) und für die Edicta-Grafik (incl. Ausschnitt-Kopie und verschiedene Druckoptionen).
INFO	.KLX	010	4.1	Olaf Krumnow, Herbert zur Nedden Anzeige der RAM 6.0-Sprünge und -Variablen, Utility-Funktionen, BDOS- und BIOS-Funktionen, diverse System-Datenstrukturen (FCB, DPH, XDPB, ENV), Systemeinsprünge à la Memotech, OK-Menü und FastScr samt Erläuterungen sowie Übergabeparameter soweit erforderlich.
KALENDER	.KLX	005	4.1	Herbert zur Nedden Halt ein Kalender, mit Blätterfunktion für den Monat und das Jahr.

R A M x.x: KLIX-Overlays

Name	Typ	Wo	RAM	Autor/Funktion
KSWEEP	.KLX	005	4.1	Olaf Krumnow Nsweep-artiges Programm, mit dem man sehen kann, was sich wo und wie groß im KCLICK-Heap tummelt. Klarerweise können die Blöcke auch umbenannt oder gar gelöscht werden.
KWECKER	.COM	003	4.1	Jan Brederke Das erste in C geschriebene KCLICK-Overlay. Hierbei handelt es sich um einen Wecker, d.h. daß das Erreichen einer eingestellten Uhrzeit wird angezeigt.
MKFFDX	.KLX	6.0	6.0	Michael Keßler, (Herbert zur Nedden) Mini-Version von FormStar (FSFDX.KLX), welches nur die gängigen Formate formatieren kann.
MFKSDX	.KLX	6.0	6.0	Michael Keßler, (Herbert zur Nedden) dto. für die SDX
MOUSE6	.KLX	010	6.0a	Herbert zur Nedden Maustreiber für die Genius-Maus GM-6000. Die Aktionen der Maus werden in entsprechende Tastaturbefehle umgesetzt, d.h. die Maus tut so, als wäre sie Bestandteil der Tastatur.
MSGTOMAC	.KLX	006	4.7	Herbert zur Nedden Ist durch MsgToZ80.KLX ersetzt.
MSGTOZ80	.KLX	006	4.1	Herbert zur Nedden Help-Anzeige zu MsgToZ80. MsgToZ80 konvertiert eine Text-Datei in einen Assembler-Source. Dieser kann entweder nur den Text als Datendefinition oder ein Programm (.COM oder .KLX) enthalten.
NWSCHIRM	.KLX	009	4.3	Jan Brederke, (Herbert zur Nedden) Mit diesem KLIX und einem geeignet installierten New-Word bzw. WordStar 4.0 kannst Du bei laufender Editor-Sitzung das Bildschirm-Format umschalten.
NOTE	.COM	4.1	4.1	Bernd Preusing Einfacher, aber dafür sehr schneller Editor. Ein zweiter Aufruf des Programmes, d.h. nach dem Laden ins KCLICK, ermöglicht den Transfer von Daten zwischen dem Editor und der Diskette. Dieses Programm ist in der jetzigen Form nicht .KLX-tauglich, da die Diskettenzugriffs-Teile in Turbo-Pascal geschrieben sind.
PLUS32	.KLX	007	4.1	Peter Würfel Anzeige, in der für die Zahlen 1 bis 94 das ASCII-Zeichen mit dem Code Zahl plus 32 steht. Dies ist praktisch, wenn es darum geht, Bildschirm-Sequenzen einzugeben, die die Zahl mit Offset 32 brauchen.
POST	.KLX	006	4.1	Herbert zur Nedden Anzeige der Portogebühren für Briefe, Drucksachen, Warensendungen und Päckchen im In- und Ausland.
RSINIT	.KLX	005	4.4	Herbert zur Nedden (Re)Initialisierung von bis zu 6 memotechartigen seriellen Schnittstellen. Es können die Baudrate sowie die Anzahl der übertragenen Bits, die Parity und der StopBits festgelegt und jederzeit geändert werden.
Z80	.KLX	008	4.1	Herbert zur Nedden Anzeige der Bit-Schiebe- und -Rotations-Befehle der Z80 (die man sich ja nicht immer merken kann).

R A M 6.0a: Software-Erweiterungen und RAM 6.0b

DiJey und Laufwerke J: - P:

(Herbert zur Nedden, 2000)

In DiJey befindet sich irgendwo folgendes Coding:

```
CP 'A'
JP C,Label
CP 'J'
JP NC,Label
```

Bei DiJey 2.03 vom 06.09.89 ist das an Adresse 29CDh, d.h. das 'J' an Adresse 29D3h. Solltest Du Wert auf die Laufwerke J: - P: legen, so muß Du dieses 'J' durch ein Kuh ('Q') ersetzen. Das geht u.a. mit folgender Kommandozeile:

```
GET 100 DIJEY.KLX;POKE 29D3 "Q;SAVE 145 DIJEY6.KLX S
```

RAM 6.0a und der Spooler und DATE

(Herbert zur Nedden, 2000)

Wenn Du über den Spooler druckst, aus Versehen der Drucker so lange keine Daten mehr entgegennimmt, daß das freundliche Blinkfenster mit der TimeOut-Meldung kommt, Du den Drucker aktivierst und dann SHIFT-BS drückst, so kassierst Du einen Seitenvorschub an einer nicht vorgesehenen Stelle. Es ist nämlich leider so, daß das zu druckende Zeichen, welches den TimeOut verursacht hat, durch das Blinkfenster in ein FormFeed gewandelt wird. Olaf und mir ist das nie aufgefallen, da unser Drucker stets genug Papier hatte.

Solltest Du versucht haben, am 8.7.90 die Uhr mit DATE auf dieses Datum zu stellen, so klappte das nicht. Dieser Unschönheit hilft dieser Patch auch ab.

Das Resultat dieses Patches heißt dann natürlich RAM 6.0b.

Hier eine ZEX-Datei zum Patchen: RAM 6.0a --> RAM 6.0b

```
GET 100 RAM60A.COM
POKE 087D 9F
POKE 2503 90 97
POKE 5ADF 96 97
POKE 9572 C5 CD 23 34 C1 C9
POKE 9578 E5 2A 3E 5B CD 9A 5D E1 C9
SAVE 298 RAM60B.COM S
DIFF RAM60B.COM RAM60A.COM /V
```

Hier der Patch: (genauer das Ergebnis des o.g. DIFF)

Rel Offset		RAM60B.COM			RAM60A.COM			Rel Offset		RAM60B.COM			RAM60A.COM		
Hex	Dec	Hex	Dec	Asc	Hex	Dec	Asc	Hex	Dec	Hex	Dec	Asc	Hex	Dec	Asc
087D	2173	9F	159	.	96	150	.	9577	38263	C9	201	I	00	0	.
2503	9475	90	144	.	23	35	#	9578	38264	E5	229	e	00	0	.
2504	9476	97	151	.	34	52	4	9579	38265	2A	42	*	00	0	.
5ADF	23263	96	150	.	9A	154	.	957A	38266	3E	62	>	00	0	.
5AE0	23264	97	151	.	5D	93	Ü	957B	38267	5B	91	Ï	00	0	.
9572	38258	C5	197	E	00	0	.	957C	38268	CD	205	M	00	0	.
9573	38259	CD	205	M	00	0	.	957D	38269	9A	154	.	00	0	.
9574	38260	23	35	#	00	0	.	957E	38270	5D	93	Ü	00	0	.
9575	38261	34	52	4	00	0	.	957F	38271	E1	225	a	00	0	.
9576	38262	C1	193	A	00	0	.	9580	38272	C9	201	I	00	0	.

Vielen Dank an Herbert Oppmann für den Hinweis und ein Fehlerdemo-Programm zum Spooler und Hartmut Traber für eine Demo-Disk für die Probleme mit dem 8.7.90.

R A M 6.0a: Software-Erweiterungen und RAM 6.0b**RAM 6.0 und Steprate von Festplatten**

(Herbert zur Nedden, 2000)

Standardmäßig steppt RAM 6.0 Festplatten mit 3 ms. Solltest Du einen größeren Geldbeutel zur Hand haben, und Dir daher eine schnellere Platte zulegen, so ist es evtl. recht sinnvoll, die Steprate anzupassen. Steppe ich die Festplatte zu langsam, ist sie evtl. etwas zu laut.

Folgende Bytes sind in RAM60.COM (auch 60a, 60b) zu finden (alles Hex):

Byte	C9	00	00	00	00	01	00
Adresse	8F76			8F7C			

In 8F7C steht die Steprate (genauer das Command Control Byte, welches die Steprate enthält). Folgende Werte kannst Du versuchen:

Wert	00	01	02	03	04	05	06	07
Steprate	3ms	10us	25us	50us	200us	70us	3ms	3ms

(ms = Millisekunden, us = Mikrosekunden). Bei den us-Stepratzen wird ein sog. gepufferter Step ausgeführt - was immer das auch ist.

Gepufferter Step

(Herbert Oppmann, 8520)

Früher durfte ein Step-Impuls nicht schneller kommen als wie die Hardware mitgemacht hat. Zu diesem Zweck mußte der Controller wissen, wie schnell das Laufwerk steppen kann (ist ja bei Floppy-Laufwerken immer noch so).

Moderne Festplatten-Laufwerke haben jedoch einen internen Zähler, der angibt, wieviel Steps in welche Richtung noch zu erledigen sind. Der Controller kann hier fast beliebig schnell die Step-Impulse rausjagen, das Laufwerk zählt zunächst mal mit ("puffert"), während es den Arm in Bewegung setzt. Wenn das Laufwerk fertig ist, meldet es das dem Controller.

Also die beiden Vorteile:

- 1) Controller muß nicht wissen, wie lahm das Laufwerk steppt
- 2) es geht schneller!

Wenn man sich den Fall vorstellt, daß nur eine Spur weitergegangen werden soll, dann braucht man Zeit zum Beschleunigen, Abbremsen, und fuer den Ausschwingvorgang. Wenn ich jedoch z.B. 100 Steps hintereinander ausführen lasse, brauche ich soviel Zeit nicht für jeden Step! Da sich die Controller jedoch nur auf eine konstante Zeit einstellen lassen, muss ich die für einen einzigen Step zugrunde legen und verschwende bei vielen Steps nacheinander viel Zeit.

Das Laufwerk jedoch weiß stets, wie schnell es im Moment steppen kann - also warum sollte es das nicht in eigener Regie tun?

MS - kann jetzt auch Daktari und Aprikose

(Herbert zur Nedden, 2000)

Herbert Oppmann hat MS einiges beigebracht. Die Neue Version MS60A.CHN ist auf KLICK.011.

Die Änderungen:

- ein Fehler weniger (in strequal)
- unterstuetzt ATARI ST
- unterstuetzt APRICOT F1

Die ganz große neue Version wird wohl noch etwas brauchen, solange Herbert Oppmann immer noch nicht genau genug versteht, wie dieses Programm im einzelnen funktioniert.

R A M 6.0: Ein paar Anmerkungen

RAM 6.0 - ein paar Worte dazu

(Peter Würfel 7262)

Was soll's ?

Da kam also eines Tages ein Päckchen mit drei Disketten und einem ganzen Berg Papier: RAM 6.0. Doch brauch ich das überhaupt? Mein System läuft doch fehlerfrei und das, was ich getan haben will, tut meine schwarze Kiste. Und außerdem, läuft denn all das, was ich bisher gebraucht habe, ohne Probleme mit dem neuen Betriebssystem?

Meine Anlage wurde von mir unter RAM 4.5 so eingerichtet, daß ich mir beim Benutzen keine Gedanken mehr machen mußte, was ich wie tue, dazu habe ich mir ein Menu.cpr mit etwa 40 Bildschirmen und über 300 Befehlszeilen geschrieben, um meine 700nochwas-k-RAM-Disk, meine drei Diskettenlaufwerke, meine EPROM-Floppy und die SRAM-Floppy 'im Griff' zu haben. Wünsche hatte ich natürlich schon noch (siehe alte Infos), aber die waren nicht so gravierend, daß mir ein neues Betriebssystem als unvermeidlich erschienen wäre. On verra. Also erstmal die RAM 6.0-Dokumentation lesen: Und da wimmelte es nur so von 3-Buchstaben-Kürzeln und erstmal war einiges schleierhaft. (Wobei die Dokumentation wirklich gut geschrieben ist! Es ist halt einfach eine ganze Menge Neues hinzugekommen und da muß man sich erstmal reindenken) Die Installation von RAM 6.0 klappte auf Anhieb, wird man doch von Herbert 'an die Hand genommen' und erhält alles, was man zu tun (oder zu lassen) hat, übersichtlich auf dem Bildschirm angezeigt.

Und dann der erste Versuch, ob das neue MENU.COM mein altes Menu.cpr 'verdaut'. Doch welch Graus, da wollte einer wohl was ganz Gutes machen und hat mal wieder eine feste Zeilenzahl in den Bildschirmteil eingebaut! Mit Moni auf die Suche gemacht, herauszufinden, wo denn die Zeilenzahl in MENU.COM steckt. Vielleicht kann man da einfach was patchen. Aber Fehlanzeige! Ein Telefonanruf bei Claudio brachte mich auch nicht weiter. (Claudio, Du hast mir versprochen, ein neues MENU.COM zu schreiben. Laß es meinetwegen bleiben! Warum? Siehe unten!) Also RAM 6.0 wieder rausgeschmissen und mein altes RAM 4.5 geladen. Für mich ist Menu das Herz meines Systems und wenn das nicht funktioniert, dann verzichte ich auch auf die anderen Features - dachte ich. Bis ich dann bei einem Telefongespräch mit Herbert den entscheidenden Tip bekam, wie ich das 'alte' Menu.com auch unter RAM 6.0 betreiben kann. Ausprobiert: Es klappt!! Und da begann dann endlich die eigentliche Auseinandersetzung mit RAM 6.0 und in der Zwischenzeit möchte ich diese Möglichkeiten nicht mehr missen. Vieles liegt für mich noch im Dunkeln, doch das, was läuft, das klappt hervorragend und peu a peu werden dann die anderen Feinheiten des neuen RAM noch erkundet, entdeckt, ausprobiert und - wenn brauchbar - angewendet werden!

Leute, RAM 6.0 ist ein Muß!

So und nun ein paar Einzelheiten:

Das NZEX von RAM 6.0 versus ZEX von RAM 4.x (Grundsätzliches)

Auf den ersten Blick (ins Handbuch) sah alles genauso aus wie unter RAM 4. Doch in der Praxis verhielt sich NZEX bei mir erstmal wie ein SUB. Im letzten Info kam dann der (entscheidende) Hinweis, der (leider) im Handbuch fehlt! Wenn man mit NZEX eine Tastatureingabe simulieren will, dann muß vor dem Programmaufruf ^% (zwei Zeichen) stehen!

Beispiel - Ich will NewWord starten und die Datei 'test.txt' ausdrucken lassen und anschließend NW wieder verlassen:

ZEX(RAM4.x):	NZEX(RAM6.x):
nw	^%nw
ptest.txt^Äx	ptest.txt^Äx

R A M 6.0: Ein paar Anmerkungen**NZEX und NewWord (Suchen und Ersetzen)**

In Info 33-15 hab' ich mich über das Problem 'NW-Suchen und Ersetzen' ausgelassen. Dort habe ich folgende Lösung gezeigt: ^PR^P und man mußte dann von Hand <CTRL><E> und <CTRL><R> eingeben. Bei NZEX muß dieser Ausdruck wie folgt aussehen: ^P^ER^P^R. Eine Benutzereingabe ist hier weiterhin notwendig, doch es reicht, wenn man zweimal die Leertaste betätigt. Die dort vorgestellte Zeile

^*^<^N<CTRL><E> u. <CTRL><R> eingeben: ^O^>^PR^R

sieht nun deshalb so aus:

^*^<^NZweimal Leertaste drücken ! ^O^>^P^ER^P^R

(^X bedeutet: zwei Zeichen; ^X bedeutet ein Zeichen, über <CTRL><P> eingegeben)

Wenn man bei NW während 'Suchen und Ersetzen' die Leertaste drückt, wird dadurch ja die Bildschirmanzeige abgeschaltet und das Suchen und Ersetzen findet dann quasi im Hintergrund (und somit schneller) statt. Dieser Leertastendruck war auch dann möglich, wenn via ZEX gesucht und ersetzt wurde. NZEX mag diesen Druck auf die Leertaste nicht, es sei denn man ersetze ^X durch ^X^X. Und da dieses doppelte Escape, das die weitere Arbeit nach dem Ersetzen ermöglicht, auch dann nicht schadet, wenn man nicht die Leertaste betätigt, kann es vorbeugend grundsätzlich in die ZEX-files gesetzt werden.

MENU und RAM 6.0

Da kam also mit RAM42 ein MENU (Version 1.5), das gegenüber den vorherigen Versionen einige Vorteile aufwies. Nur, dann stellte sich heraus, daß dieses Menu nicht mehr als 16 k Menu.Cpr verarbeiten konnte. Eine Anfrage in den Infos brachte Hans Gras auf den Plan, und der schickte mir eine Version 1.62. Diese verkräftete endlich mehr als 16 kByte und funktioniert(e) bei mir fehlerfrei. Ich hab durch Patchen diese Version an meine Bedürfnisse angepaßt und fand die Sache sei erledigt, bis ich eines Tages meine Erfahrungen mit Menu in Hinblick auf RAM 5 für Herbert zusammenschreiben wollte und dabei feststellte, daß diese Version 1.62 von Hans gar nicht veröffentlicht wurde, sondern nur eine Version 1.63. Und diese Version ist einerseits zwar verbessert, aber was heißt schon "besser"? Die Version 1.63 erkennt nämlich (so schließe ich wenigstens aus dem, wie sich das Programm verhält), welches Bildschirmformat gerade aktuell ist und gibt dementsprechend die Aufforderung zur Eingabe aus. Dies mag zwar praktisch erscheinen, mir paßt das jedoch aus zwei Gründen nicht.

- 1) Wenn ich z.B. in SuperCalc gerade im 56-Zeilen-Bildschirm bin, kann ich direkt aussteigen, denn dann stellt mir mein Menu 1.62 den Bildschirm wieder auf das Standard-Format 28 Zeilen zurück, da bei mir in der ersten Zeile des Bildschirms '^XAG1' steht. Bei Menu 1.63 würde die Aufforderung zur Eingabe an 'unmöglicher' Stelle stehen, es sei denn der 56-Zeilen-Bildschirm bliebe auch in Menu erhalten.
- 2) Die fest eingestellte Zeilenzahl in Menu 1.63 nimmt mir die Möglichkeit, die Window-Routinen von RAM 4.2ff zu nutzen. Wenn ich einen Bildschirm immer 'am Stück' aufbauen muß, ist eine Änderung sehr viel mühsamer, wie wenn ich den Bildschirm in einzelne Windows zerlegen kann. (Mein Hauptmenü, so wie es im Info 34-1 abgedruckt war, setzt sich z.B. aus 8 Windows zusammen.) Also hab ich die Finger von Menu163 weggelassen und bin bei der Version 1.62 geblieben.

Menu Vers. 1.62 wurde mir so gepatched, daß der Zeilenzähler im Bildschirmaufbau quasi 'ausgeschaltet' ist. (Die Bildschirmgröße wurde auf 255 Zeilen gesetzt) Damit können die Window-Steuerbefehle des RAM- Bildschirmtreibers verwendet werden.

R A M 6.0: Ein paar Anmerkungen

Damit dieses Menu unter RAM 6.0 läuft, wurde mit GENINS die Adresse der MCL von RAM 6.0 installiert. Diese Version heißt nun 1.62a, Ihr findet sie zusammen mit einem kurzen Beispiel-MENU.CPR und Hinweisen zum Bildschirmaufbau auf KLICK.011.

Um zu verstehen, warum ich mit dieser Version im Zusammenspiel mit RAM 6.0 so happy bin, hier ein Auszug aus meiner Wunschliste zu einem neuen MENU, die ich zu Beginn des Jahres an Herbert und Claudio Romanazzi schickte:

- 1) Ich hätte gerne die Möglichkeit, daß in einer MENU.CPR-Befehlszeile auch nach einem "-"Zeichen noch Befehle folgen können. Es müßte dann also folgende Befehlszeile möglich sein:

```
Aa;:xdir "Suchmaske ;i:
```

Also: wenn ich 'A' wähle, soll Laufwerk A: eingelogged werden, das Directory mit der gewünschten Maske angezeigt und dann anschließend wieder Laufwerk I: eingeloggt werden. (Damit Menu.com sein Menu.cpr, das bei mir auf I: liegt, auch wieder findet)

- 2) Ein weiterer Wunsch dürfte, wenn überhaupt möglich, wohl nur schwer zu erfüllen sein. Ich erkläre es am besten an einem Beispiel. In MENU.CPR steht z.B. folgende Befehlszeile:

```
Axdir a:*.bas;ebasic "Compilier-Datei
```

Diese Zeile wird von MENU16 wie folgt abgearbeitet: MENU verlangt sofort meine Eingabe und arbeitet erst dann die Befehlszeile ab.

Ich möchte nun gerne folgenden Ablauf: Wenn ich Option 'A' wähle, startet MENU den ersten Befehl, es wird mir also das Directory von A: mit der Maske '*.bas' angezeigt. Erst jetzt soll mich MENU zur Eingabe des Dateinamens auffordern, der zum Aufruf ebasic... gebraucht wird.

- 3) Und die folgende gewünschte Möglichkeit wäre dann 'der Gipfel', eine Befehlszeile die so aussieht:

```
Axdir a:*.bas;ebasic "Compilier-Datei ;erun a:"Was starten?"
```

Eine Befehlszeile also, die mir die Möglichkeit gibt, zwei Eingaben in einer Befehlszeile vom Bediener zu verlangen.

- 4) Hans Gras hat auf CLUB.032 ein Programmle namens 'IF.COM', das die Fortsetzung bzw. den Abbruch eines Sub-files veranlaßt, je nachdem, ob eine bestimmte Datei vorhanden ist. Könnte man sowas nicht in Menu einbauen, damit dann z.B. folgende Befehlszeile möglich ist:

```
a:if a:test.com (mdc c:=a:test.com;test)test
```

Das würde dann bedeuten: Wenn auf Laufwerk A: das file 'test.com' nicht vorhanden ist, dann werden die Befehle, die in Klammern folgen in die Befehlszeile übernommen, wenn 'test.com' vorhanden ist, werden die Befehle, die auf die Klammern folgen ausgeführt.

Soweit die Auszüge aus dem damaligen Brief. Und nun das Tolle: All das (und noch mehr) klappt mit Menu 1.62a und den Möglichkeiten von RAM 6.0 !!!

Wie ? - So : (nämlich im Zusammenspiel mit ALIAS.CMD)

Beispiel 1 (Wunsch 1):

In der Menu-Befehlszeile steht:

```
AXDIRX "Suchmaske :
```

Und die entsprechende ALIAS-Zeile lautet dann:

```
XDIRX a;:xdir $1;i;:
```

R A M 6.0: Ein paar Anmerkungen

Beispiel 2 (Wunsch 2):

In der Menu-Befehlszeile steht:

Axdir a:*.bas;EBASX

Und die entsprechende ALIAS-Zeile lautet dann:

\$"Compiler-Datei : "ebasic \$'L1

Beispiel 3 (Wunsch 3 und noch mehr!)

In diesem Beispiel habe ich nicht das oben gewünschte Befehlszeilen-Beispiel darstellen, sondern einen anderen, noch komplizierteren Fall:

Was möchte ich?

Ich möchte aus einer Library, die sich auf einem beliebigen Laufwerk befindet, mittels Lbrext.com bestimmte Dateien extrahieren und zwar auf ein anderes Laufwerk und außerdem mit den Optionen U (entkomprimieren) und O (überschreiben).

In der Befehlszeile in Menu.cpr steht:

ALBRETXO UO "In welchem Laufwerk ist die Library ? <LW-Buchstabe>:

Menu erfragt nun zuerst das Laufwerk, in dem die Library zu finden ist.

In ALIAS.CMD brauche ich nun 3 Zeilen:

LBRETXO xdir \$2?:*.lbr;LBRETXO1 \$1 \$2

Diese Alias-Zeile bewirkt, daß mir nun erstmal alle Libraries angezeigt werden, die sich auf der Diskette auf diesem Laufwerk befinden

LBRETXO1 \$"^G^J<User> \$"^G<Name>(ohne .lbr) "echo ^AA;

lx ldir \$2\$L1:\$'L2;echo ^AS;LBRETXO2 \$1 \$2\$L1 \$'L2;

Diese nächste Alias-Zeile erfragt nun, aus welcher Library ich Dateien extrahieren möchte und zeigt mir anschließend den Inhalt dieser Library.

LBRETXO2 \$"^G<Ziel> (dir:afn1,dir:afn2,..)"lx lbrext \$2:\$3 \$'L1 \$1

Und nun kann ich eingeben, welche Datei ich wohin haben möchte (und anschließend wird's getan!)

Also, Wenn das nix is !?!

Und zu Wunsch 4 brauch ich ja nichts mehr zu sagen, in Menu gibts zwar kein IF, dafür aber in Alias.

Kann mir nun jemand noch sagen, wieso ich ein neues Menu bräuchte?

Anm.d.HzN: Das von Peter erwähnte IF ist übrigens kein Feature der Aliase sondern des Flow Command Package. Daher kann ich IF in allen Kommandozeilen verwenden - auch unter SUB und ZEX.

R A M 6.0: FestplatteEiniges zum Thema Festplatten

(Holger Hansen, 3300)

Im Laufe der letzten beiden Monate haben ich einige Anfragen zum Thema Festplatte und RAM 6 erhalten. Das folgende ist eine Zusammenfassung der Fragen und der Antworten.

Bug (=Wanze) in HDIO.INC:

Mir ist leider ein peinlicher kleiner Fehler im Programm Hdserv unterlaufen. In der Datei HDIO.INC ist in der Procedur Format_The_Drive steht bisher

```
...
Track:=1;
```

es muss aber heißen

```
...
Track:=0;
```

Es führt dazu, daß beim Formatieren bisher Track Null nicht mit formatiert wurde, was zu Problemen bei Rcheck6 führt. Bevor ihr nun neu kompiliert, lest erstmal die Änderungen weiter unten.

Einstellung auf dem Controller

Jumper	OMTI		Beschreibung
	5510	5520	
W1	X	X	Wird für die Controllerkennung gebraucht. Sonst keinen Zweck an ECU Rechnern
W2	0	0	
W3	X	X	
W4	0	0	
W5	0	0	Basis Portadresse (hier Basis+8)
W6	X	X	(also für FDX und ECB Adapter)
W7	X	X	PC Bios abschalten, weil MTX keins hat
W8	0	0	Sitz des PC-Bios (uns egal siehe W7)
W9	X	*	beim 5510 Sektorgröße (1024 *9)
W10	X	*	beim 5520 nicht belegt
W11	0	*	Soft-Sektor
W12	0	*	Plattentyp (Fest), (Wechselplatten
W13	0	*	haben hier jeweils Steckbrücken)
W14	0	*	W11 für die zweite Platte
W15	0	*	W12/W13 für die zweite Platte
W16	0	*	

* heißt beim OMTI 5520 nicht genutzt

Unterschiede zwischen 5510 und 5520 sind die unterschiedliche Handhabung der Sektorgröße und der Zahl der Sektoren pro Track. Der 5520 ist fest auf das MSDOS Format 17 Sektoren mit je 512 Bytes pro Track eingestellt. Der 5510 ist hier flexibler. Er beherrscht 4 unterschiedliche Größen. Vorteil ist die höhere Geschwindigkeit und die größere Speicherkapazität.

9*1024 = 9 kBytes pro Track

17*512 = 8,5 kBytes pro Track

das macht bei 614 Cylindern und 4 Köpfen 1228 kBytes mehr Speicher.

R A M 6.0: Festplatte

Softwareseitig: HDSERV

Anmerkung:

Physikalischer Track: eine Festplattenspur geht über eine Seite
 Physikalischer Sektor: ein Sektor auf der Festplatte (512 oder 1024) Bytes
 Logischer Track: Track wie er vom Betriebssystem verwaltet wird
 (hier geht er über alle (4) Seiten)
 Logischer Sektor: CP/M's Sektorgröße ist 128 Bytes die Umrechnung auf
 phys. Sektoren geschieht im BIOS (sprich RAM 6.0)

	OMTI 5510	OMTI 5520 (+5510 auf 17*512)	
	(9*1024 Bytes)	(17*512 Bytes)	
Sectorlaenge =	1023	511	physikalische Sektorlaenge - 1
PortBasis =	\$48	\$48	für FDX Slot /* Steckbrücke W5/W6 */
	\$E8	\$E8	für EDBus
RpPT =	72	68	Zahl der 128 Byte Records/ Track
SpPT =	9	17	phys. Sektoren pro Track
RpPS =	8	4	Zahl der 128 Byte Records /phys. Sektor
MaxHead =	4	4	Zahl der Köpfe
MaxCyl =	614	614	Zahl der Cylinder-1
MaxSec =	288	272	RpPT * MaxHead
LUN =	1	1	Phys. Festplatte Nr. 1
	2	2	" " " 2

Drive_Characteristics : array [0..7] of byte =
 (2, Highbyte MaxCyl+1 (614 = \$266)
 \$66, Lowbyte Maxcyl+1
 4, MaxHead
 0, HighByte für Start Track Reduced Write Current
 0, Low-Byte für Start Track Reduced Write Current
 1, HighByte für Start Track Write Precompensation
 0, Low-Byte für Start Track Write Precompensation
 0) Abschließende Null

Die Werte für Reduced Write Current und Write Precompensation muß man dem Festplattenbeipack-Zettel entnehmen. Die Werte oben gelten für z.B.

Seagate ST 225, BASF 6188 R3

Im Manual für den OMTI sind Werte für einige gebräuchliche Platten aus der MSDOS Welt.

In der bei RAM6 mitgelieferten Datei HDSERV.CHN gelten die folgenden Werte:

17 * 512 Bytes
 Port \$E8
 obige Drive_Characteristics

Softwareseitig: FORMATE.R6 (bearbeiten mit FMT6)

Es muß ein Format erstellt werden:

Bspl.: 20 MB BASF 6188 mit Omti 5510, 615 Tracks (in Klammern Werte für 5520) installiert auf dem EDBus

wird in 4 logische Laufwerk J,K,L,M aufgeteilt
 J 150 Tracks , 1 Systemspur, 512 Direintraege
 K 300 Tracks , 150 Systemspuren, "
 L 450 Tracks , 300 Systemspuren, "
 M 615 Tracks , 450 Systemspuren, "

(Anmerkung: Mein Bootlaufwerk ist I:, also alle Angaben die das Bootlaufwerk betreffen entsprechend abändern.) Ihr braucht für 2 Festplatten (physik.) 2 unterschiedliche Formatbeschreibungen. Die sich aber nur in der Festplattennummer unterscheiden müssen.

R A M 6.0: Festplatte

Die folgenden Meldungen sind abgekürzt, weil ich keine Lust hatte selbige komplett abzupinseln.

Formate:

<pre>I>fmt6 E editieren CfigCode <FF> 30 Ä für Ändern Format-Name : Hard-J Format-Typ : 2 Controller : E phys. Sektor: 1024 (512) ph.Sekt/Spur: 9 (17) Cylinder : 150 Köpfe : 4 1. Sektor : 0 Skew : J Harddisk : 1 Systemspuren: 1 Blockgröße : 4 Dir-Einträge: 512 Z zurück CfigCode <FF> 31 K Kopieren CfigCode : 30 Ä für Ändern Format-Name : Hard-K Format-Typ : Ret Controller : Ret phys. Sektor: Ret ph.Sekt/Spur: Ret Cylinder : 300 Köpfe : Ret 1. Sektor : Ret Skew : J Harddisk : Ret Systemspuren: 150 Blockgröße : Ret Dir-Einträge: Ret Z zurück</pre>	<pre>CfigCode <FF> 32 K Kopieren CfigCode : 30 Ä für Ändern Format-Name : Hard-L Format-Typ : Ret Controller : Ret phys. Sektor: Ret ph.Sekt/Spur: Ret Cylinder : 450 Köpfe : Ret 1. Sektor : Ret Skew : J Harddisk : Ret Systemspuren: 300 Blockgröße : Ret Dir-Einträge: Ret Z zurück CfigCode <FF> 33 K Kopieren CfigCode : 30 Ä für Ändern Format-Name : Hard-M Format-Typ : Ret Controller : Ret phys. Sektor: Ret ph.Sekt/Spur: Ret Cylinder : 614 , (s.u.) Köpfe : Ret 1. Sektor : Ret Skew : J Harddisk : Ret Systemspuren: 450 Blockgröße : Ret Dir-Einträge: Ret Z zurück</pre>
---	---

Anmerkung: zu Config-Code 33: Cylinder = 614, weil Zylinder 615 fürs Parken reserviert ist. Wer die 4*32 kBytes unbedingt braucht, kann hier 615 eintragen

<pre>CfigCode <FF> FF Configurieren C H Hinzufügen CfigCode :30 H Hinzufügen CfigCode :31 H Hinzufügen CfigCode :32 H Hinzufügen CfigCode :33 S Sichern.. erzeugt RAM6.FMT Z Zurück B Bootsektor editieren Laufwerk : I (mein Bootlaufwerk) C Configs ändern</pre>	<pre>Laufwerk J Config 30 C Configs ändern Laufwerk K Config 31 C Configs ändern Laufwerk L Config 32 C Configs ändern Laufwerk M Config 33 Z Zurück Sichern Laufwerk : I X Ende</pre>
--	--

R A M 6.0: FestplatteSoftwareseitig: Änderungen in RAM 6

Inst60 Ram60

F Formate aus RAM6.FMT einlesen

H Hardware der Laufwerke

alle Fragen zu Diskettenlaufwerke mit Ret beantworten

...

Anzahl der Festplatten: 1

Controller : E

Cylinder : 615 hier der richtige Wert zum Parken

Köpfe : 4

Reduced Write Current : 0

Write Precompensation : 256

...

(Ramflopies Fragen mit Ret beantworten)

X Ende

J Sichern

Nach dem Booten müßte nun die Festplatte für das System vorhanden sein. Mit HDSERV FORMAT 4 kann man nun die Festplatte formatieren (da HDSERV den Controller direkt ansteuert, braucht die Festplatte noch nicht ins System eingebunden zu sein, nur elektrisch)

Softwareseitig: Bad Sektors

Auf den meisten Festplatten ist eine Liste der defekten Sektoren auf der Platte aufgedruckt. Vor allem bei alten Platten kann diese Liste schon mal etwas länger sein. Diese Liste hat meist folgenden Aufbau:

Cyl	Head	Bytes
23	3	765
372	1	4670

.. .

dies bedeutet, daß der Hersteller seine Garantie für die Festplatte nicht an diesen Stellen geben kann. Das heißt nicht, daß ein Programm wie RCHECK hier einen defekten Sektor anzeigt. Unter Umständen kann man diese Stelle ein paar hundertmal lesen und schreiben. Trotzdem sollte man diese Stellen auf der Platte als defekt markieren. Anders als bei MSDOS Rechnern ist aber bei CPM Rechnern keine einheitliche Vorgehensweise oder gar Dateiattribut für defekte Plattensektoren vorgesehen. Man behilft sich folgendermaßen:

- 1) Taschenrechner zücken (Calc.klx tut es auch)
- 2) Umrechnen Bytes in logische Sektoren auf einer Seite
- 3) Umrechnen von Head in logischen Sektor, dann mit 2) addieren
- 4) feststellen in welcher Partition der Cyl steht.
- 5) Auf der entsprechenden Partition ein File BADSECT.OR mittels SAVE 0 BADSECT.OR erzeugen
- 6) Aufruf von DU3
- 7) mit txx den Track eingeben (xx=23 im Bspl.)
- 8) mit sxx den logischen Sektor eingeben.
- 9) jetzt zeigt DU3 nach einem D (=DUMP) die GROUP oder Block an, bei dem der Fehler liegt. Group (Zahl in HEX vor dem Doppelpunkt reicht) notieren
- 10) nächste defekte Stelle behandeln. bei mehr als 255 Blöcken pro Partition (üblich bei Harddisk) passen 8 Blöcke in einen Eintrag (Achtung 16 Bit Wert erst Low dann High Byte!!!)

R A M 6.0: Festplatte

Beispiel:

4 Partitionen J 0 .. 150
 K 151 .. 300
 L 301 .. 450
 M 451 .. 615

9 phys. Sektoren mit je 1024 Bytes
 logische Sektoren = Sektor mit 128 Bytes
 logisch 288 Sektoren pro Track, also 72 Sektoren pro Kopf
 erster logischer Sektor hat Nummer 1

Fehler: Cyl. Head Bytes
 23 3 765
 372 1 4670

- 1) erledigt
- 2) 765 Bytes div 128 = 5 => 5. logischer Sektor
- 3) Head 3: => 3 * 72 Sektoren = 216 Sektoren
 + 5 Sektoren
 ergibt: 221 Sektoren

- 4) Cyl 23 => also Laufwerk J
- 5) save 0 j31:BADSECT.OR
- 6) DU3 LJ
- 7) T23
- 8) S221
- 9) D

G0047:04 T23 S221 P221

- 10) mit Fbadsect.or
 D *zeigt DU3 nun den Verzeichniseintrag von BADSECT.OR*
 40 1F424144534543544F52000000000000
 50 00000000000000000000000000000000
 ch4F 10 *Pro Group 10h aufaddieren*
 ch50 47 00
 40 1F424144534543544F52000000000010
 50 47000000000000000000000000000000
 w *schreibt die Änderungen auf Platte*
 mit Fehlern auf der selben logischen Platte genauso verfahren. Z.B. weitere Fehler auf Laufwerk J in G0105:00 müßte so eingetragen werden:
 ch4F 20 *Pro Group 10h aufaddieren*
 ch52 05 01
 D
 40 1F424144534543544F52000000000020
 50 47000501000000000000000000000000
 w
 Gruppen bei BADSECT.OR mit eintragen. Verfahrensbedingt geht hier immer ein ganzer Block flöten, auch wenn nur ein Sektor betroffen ist.

- 2) 4670 Bytes div 128 = 36 => 36. logischer Sektor
- 3) Head 1: => 1 * 72 Sektoren = 72 Sektoren
 + 36 Sektoren
 ergibt: 108 Sektoren

- 4) Cyl 372 => also Laufwerk L
- 5) SAVE 0 L31:BADSECT.OR
- 6) DU3 LL
- 7) T372
- 8) S108
- 9) D

G0251:04 T372 S108 P108

- 10) mit FBADSECT.OR
 D *zeigt DU3 nun den Verzeichniseintrag von BADSECT.OR*
 40 1F424144534543544F52000000000000
 50 00000000000000000000000000000000
 ch4F 10
 ch50 51 02
 40 1F424144534543544F52000000000010
 50 51020000000000000000000000000000
 w *schreibt die Änderungen auf Platte*

L e s e r b r i e f: Hans Gras, NL-1506

Diesmal etwas über meine Harddisk Geschichte.

Meine Harddisk (TANDON TM 503, 15MB, 306 Cylinder) läuft seit mehreren Wochen. Aber es hat etwas Zeit gekostet, so weit zu kommen. Einige Sachen sind bereits im Info vermerkt, aber das wußte ich vor vielen Monaten nicht. Die Fehler, die ich gemacht habe werden nicht vermerkt in dieser Geschichte. Nur die (hoffentlich) nützlichen Informationen. Dank an Holger Hansen und Claudio Romanazzi für ihre Hilfe.

- Als erstes habe ich RAM 6.0 zum laufen gebracht.
HD nicht sofort aber später Installieren! Erst mal sehen was RAM 6 uns bringt.
- Ich habe von Hannes Mehrtens einen OMTI Controller bekommen, weil in Holland nur Fremd-Hersteller Platinen zu kaufen sind. Es müßte mit diesen Platinen auch funktionieren, aber keine Garantie
- Von Gerhard Witzel habe ich die FDX/OMTI Adapter bekommen.
Vielen Dank Gerhard, für die Lötarbeit des Verdratens.
- Auf diese Platine verschiedene Teile (IC's, Widerstände usw.) gelötet (auch für die Uhr).
- Ein schwierige Frage war: Benötigt der OMTI 5520A 12V oder nicht?
Handbuch war nicht 100% klar. Ich habe die Platine mal angesehen und gesehen, daß die 12V Anschlüsse Leiterbahnen haben. Daraus habe ich verstanden: 12V schadet nicht. So habe ich die 12V auch verdratet.
- Test-Versuch: Adapter+OMTI in die FDX gesteckt, Booten, keine HangUps.
- Die HardDisk aus einem Televideo TS802H geholt, die Kabel auch.
- HardDisk auf das TVI Netzteil und OMTI angeschlossen. Kabel waren bereits vorhanden.
- Wieder testen. HD läuft an, kein Problemen.
- Dann etwas wichtiges: vom HDSERV-Paket HDIO anpassen für HD, und in HDSERV.PAS die Procedure Formatieren ändern TRACK=0 (war ein BUG). Neu Compilieren.
- Mit HDSERV FORMAT 4 die ganze HD formatieren. Hat bei mir 2'04" gedauert.
- Wenn alles geklappt hat:
RAM 6.0 neu installieren auf eine 2. Diskette für das HD.
Deine alte Bootdisk nicht wegschmeißen!
Sektoren fangen mit 0 an! Blockgröße ist 4KB. Ich habe mit 2KB angefangen, bringt aber kaum Vorteile. Ist allerdings langsamer als 4BK Blöcke.
Ich habe alle Laufwerke von J: bis P: benutzt.
In 7 Wochen habe ich 4x die verteilung geändert!
Z.B.: Laufwerk P: war zu klein um alle .HLP Dateien zu behalten. Oder Laufwerk mit ZCPR 3.3 Files hat zu wenig DIR-Einträge. Und mein (Fat)Catalog
Laufwerk L: war zu klein geworden, weil ich CLUB.8?? und CLUB.9?? auch habe.
- Eine HD ist schneller als eine Floppy, aber das ist wohl klar.

Viel Erfolg bei Deiner HD Installation.

Weitere Anmerkungen betr. RAM 6.x und/oder ZCPR 3.3:

- Automatische Funktions-Taste nach Klick Warmboot:
Ist mir nicht gelungen, ich habe jede Funktionstaste der Tabelle 1 mit etwas belegt (1^m...32^m, und mal angesehen was gestartet wird. Keiner wird gestartet. Meistens kommt ein '§' oder Control irgendwas.
Anm.d.HzN: Mit INST60 unter 'Dieses und Jenes' muß installiert werden, welche F-Taste beim KLICK-WarmBoot aktiviert werden soll. Nach Reset ist das dann auch der Fall - die Taste selbst sollte auch belegt werden.

Leserbrief: Hans Gras, NL-1506

- Warum werden die RS-232 Schnittstellen nicht initialisiert beim Hochfahren von RAM 6.0 (wie beim RAM 4.x)?
Anm.d.HzN: Dafür gibt es RsInit.KLX oder die Init-Routine im I/O-Package. RAM 6.0 ist so schon sehr lang und daher flog dieses Teil da raus, zumal viele Programme, die mit der RS232 arbeiten, diese eh selbst initialisieren.
- Ist es möglich im BIOS für die HD einige Fehler Meldungen zu bekommen. Ich habe am Anfang mehrere Abstürze gehabt - ohne Erklärung warum. (Später wußte ich warum. Track 0 war zB. nicht formatiert!)
Anm.d.HzN: Das BIOS meldet in der Tat bestimmte Fehler, wenn der OMTI diese an RAM 6.0 meldet. Nur konnte ich das nicht so recht testen - ich habe keine defekte HD (nicht mal eine heile!)
- Eine Idee: Automatischer Park der HD-Köpfe, wenn 5 Minuten nichts auf der HD passiert ist. Nicht zusammenbauen mit SchirmSaver, weil wir zB mit ZEX Backups machen, und alles läuft in Quiet Modus.
Anm.d.HzN: Keine schlechte Idee.
- Ein Hinweis: In der TCAP ist ein TINIT und DINIT vorgesehen. Das ist nützlich, um zB. die Uhr Ein- und Aus zu schalten.
Einige Programmen, zB. SHOW, QL benutzen das.
- Mit dem neuen RAM 6.0a ist eine bereits vermerkte Unschönheit nicht gelöscht: Cursor mit zwei Streifen übereinander.
Anm.d.HzN: Ich weiß von diesem Fehler. Nur kann ich ihn bei mir nicht erzeugen. Wer kann, schicke mir bitte eine Diskette von der ich booten und den Fehler nachvollziehen kann (d.h. mit einer Anleitung, welche Tasten ich drücken muß). Daher ist dieser Fehler auch noch in RAM 6.0b.
- Geschwindigkeit 03, 09, 1A, 1D (und vielleicht mehrere):
Ich habe mal einige Stunden experimentiert mit verschiedenen Interleaves. Folgendes ist rausgekommen:

Wenn es (überhaupt) jemanden interessiert, um mal auf 4Mhz zu ändern: Die auf der nächsten Seite gelistete Datei als HEX assemblieren, und mit MLOAD über das vorhandene FORM6.COM laden. Achtung: Diese Ergebnis ist Optimal für ACPY.COM. Andere Programme können deutlich langsamer sein. Wenn man genannte Formate für Arbeits-Disketten verwendet, ist es nützlich, um zu testen, was i.a. schneller ist. Ich habe eine Harddisk; Disketten benutze ich meistens um hin und her zu kopieren. Und dafür habe ich ACPY.COM.

BenchMark: Test 260KB von RAMDISK-->Floppy:

Typ	Alt Ohne Verify	Alt Mit Verify	Neu Ohne Verify	Neu Mit Verify
03	1'25"	4'42"	1'27"	2'35"
09	1'04"	2'37"	1'10"	2'12"
1A	1'16"	2'31"	1'07"	2'15"
1D	1'27"	2'51"	1'07"	2'21"

Siehe auch das letzte Info über dieses Thema. Auch RAM 6.0a läuft bei mir nicht ganz schnell.

Anmerkung für Herbert:

Vielleicht können diese Interleave Tabellen gewählt werden, abhängig ob man mit 4 oder 8Mhz arbeitet. Geschwindigkeit ist in TCAP vorhanden (und in RAM 6 auch), so sollte kein großes Problem sein, etwas mehr Arbeit in FORM6 zu stecken.

Anm.d.HzN: Grundsätzlich: JA

L e s e r b r i e f: Hans Gras, NL-1506

```

TITLE   FORM61.MAC
SUBTTL  With modified interleave tables for 03, 09, 1A, 1D.
.Z80

CFIQ_TBL:
ORG     OCDDh
DB      0B1h,0F8h,-1
DW      IL1,IL2
DB      09h,0Ah,0FBh,-1      ; <-- Format 09
;       DW      IL1,IL2      ; --> Original
DW      IL3,IL3      ; <-- New Interleave
DB      0Ch,-1
DW      IL4,IL5
DB      0B0h,-1
DW      IL7,IL7
DB      19h,1Ah,1Bh,-1      ; <-- Format 1A
;       DW      IL8,IL9      ; --> Original
DW      IL8,IL8      ; <-- New Interleave
DB      0B3h,-1
DW      IL8,IL8
DB      0F1h,-1
DW      IL10,IL10
DB      0Fh,-1
DW      IL11,IL12
DB      1Ch,1Dh,-1      ; <-- Format 1D
;       DW      IL13,IL14     ; --> Original
DW      IL13,IL13     ; <-- New Interleave
DB      03h,07h,0Bh,0Dh,-1 ; <-- Format 03
DW      IL16,IL17
DB      08h,0Eh,-1
DW      IL19,IL19
DB      11h,12h,13h,-1
DW      IL20,IL20
DB      -1

ORG     OD56h
IL1:    DB      1,4,2,5,3,0
IL2:    DB      1,4,2,5,3,0
;IL3:   DB      5,3,1,4,2,0      ; <-- Original: not used
IL3:    DB      1,3,5,2,4,0      ; <-- For 4Mhz: Interleave 3
IL4:    DB      1,2,3,4,5,6,7,8,0
IL5:    DB      5,6,7,8,1,2,3,4,0
IL6:    DB      1,2,3,4,5,6,7,8,9,0
IL7:    DB      1,6,2,7,3,8,4,9,5,0
;IL8:   DB      1,4,7,2,5,8,3,6,9,0 ; <-- Original: Interleave 3
IL8:    DB      1,3,5,7,9,2,4,6,8,0 ; <-- For 4Mhz: Interleave 5
IL9:    DB      8,3,6,9,1,4,7,2,5,0 ; <-- Original: Interleave 3
IL10:   DB      1,2,3,4,5,6,7,8,9,10,0
IL11:   DB      1,8,5,2,9,6,3,10,7,4,0
IL12:   DB      6,3,10,7,4,1,8,5,2,9,0
;IL13:  DB      1,5,9,2,6,10,3,7,11,4,8,0 ; <-- Original: Interleave 3
IL13:   DB      1,10,8,6,4,2,11,9,7,5,3,0 ; <-- For 4Mhz: Interleave 5
IL14:   DB      10,3,7,11,4,8,1,5,9,2,6,0 ; <-- Original: Interleave 3
IL15:   DB      1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,0
;IL16:  DB      1,9,2,10,3,11,4,12,5,13,6,14,7,15,8,16,0 ; <-- Original: Interleave 2
IL16:   DB      1,12,7,2,13,8,3,14,9,4,15,10,5,16,11,6,0 ; <-- For 4Mhz: Interleave 3
;IL17:  DB      16,1,9,2,10,3,11,4,12,5,13,6,14,7,15,8,0 ; <-- Original: Interleave 2
IL17:   DB      9,4,15,10,5,16,11,6,1,12,7,2,13,8,3,14,0 ; <-- For 4Mhz: Interleave 3
IL18:   DB      1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,0
IL19:   DB      1,4,7,10,13,16,2,5,8,11,14,17,3,6,9,12,15,18,0
IL20:   DB      1,10,19,2,11,20,3,12,21,4,13,22,5,14,23,6,15,24,7,16,25,8,17,26,9,18,0

END

```

Anm.d.HzN: Für Experimentierfreudige werde ich den o.g. Auszug aus FORM6 auf KLICK.011 packen - allerdings unter Verwendung meiner Labels, die etwas sprechender angeben, was für ein Skew-Table gemeint ist.

L e s e r b r i e f: Hans Gras, NL-1506

- Sind in Deiner Sammlung mehrere HELPxx.LBR oder HELPxx.ARK Dateien. Mehrere fehlen, zB. SUB.HLP, ARUNZ.HLP (ohne 1 oder 2)
Anm.d.HzN: Alles, was ich habe, ist auf PD's.
- Anbei Z33FCP11.MAC als ersatz für Z33FCP.MAC. Darin war ein Fehlerchen. Wer hat dieses Fehler gemacht? Fehler war nur auf der mit RAM 6.0 gelieferte Diskette CLUB.900, aber nicht auf CLUB.903.
Ist vielleicht etwas für unsere Amerikaner?
Anm.d.HzN: Danke, ja.
- Anbei neue PATH.*. Das alte Version 3.0 (von ZCPR 3.0) hat einen Fehler. Es testet nicht auf Path Länge. Ich habe auch einige VLIB Features eingebaut.
- Auch eine DUMPDIR.* Satz. Dies ist für Z3 Environment. Eventuelle Info aus dem Source entnehmen.

So, Ende

Grüßen aus Zaandam

Hans Gras

Anm.d.HzN: Die von Hans Gras angekündigten Programme sind auf KLICK.011 bzw. CLUB.910 zu finden.

Leserbrief: Holger Hansen, 3300

33 BS, 06.07.90

Hallo Herbert

1. SHOW (auch Version 1.4) hat eine Macke bei der Anzeige der Flowcommands. Bei mir produziert es dann nur Bildschirmmüll.
Es schreibt dabei anscheinend einen Speicherbereich kommentarlos auf den Bildschirm.
Anm.d.HzN: Gelöst mit Z33FCP11 von Hans Gras, KCLICK.011.
2. Anbei eine Programmbesprechung zu einer neuen Programmiersprache, die ich mitte Mai aus Berlin erhalten habe. Ich glaube Du wirst schon die richtige Rubrik im Info finden.
3. Übrigens hatte ich einen CRC-Error im Inhaltsverzeichnis von 905. Gott sei Dank aber am Ende des Directories.
Wie kriegt man eine solche Diskette kopiert?
Lösung nach einem mittleren Wutanfall:
Da bei jeder Direktoory Schreib- oder Leseaktion das gesamte Inhaltsverzeichnis gelesen wird, funktionierte leider kein Kopierprogramm. Lösung: Mit DU3 anschauen, wieviele Inhaltsverzeichniseinträge man noch lesen kann (bei mir 96 von 128). FMT6 anwerfen, eine leeres Format suchen, Format (in meinem Fall 0A) dorthin kopieren, mit (S)chweineformat den Wert für DRM (Direktooryeinträge-1) von 7F auf 5F (=96-1) heruntersetzen, Format sichern, ins laufende RAM hinzufügen und dann, nach Configuration, mit einem Kopierprogramm die Dateien herunterkopieren. Eventuelle Dateieinträge nach dem defekten Sektor müssen dann leider mit DU3 per Hand heruntergeholt werden.
Anschließend vernichtet man die Defekte Diskette, um ihr solche Flausen einfürallemal auszutreiben.
Anm.d.HzN: Alternative: Cache! Der muß dann nur auf Auto oder Fest stehen.
4. Könntest Du mal für alle Nichttechniker ausführen, wie man einen Port erstellt? Z.B. auf einer ECB Karte möchte ich einen Port 88h ansprechen. Welche Bausteine muß ich wie verdrahten um dies dem Memotech beizupflücken.
Anm.d.HzN: Nehme ich in Angriff.
5. Patch
für DIJEY.k1x V 2.03 6.09.89
Adresse 29D2 vorher FE 4A neu FE 51
für E45.k1x
Adresse 1264 dito
Resultat: Nun kann man auch auf Laufwerk J bis P einloggen.
Dieser Hinweis gilt bestimmt auch fuer einen Haufen anderer Programme, die bei I: Schluss machen.
Man sucht mit einem Debugger (DDTZ,MONI,PATCH...) nach der Zeichenfolge FE 41 FE 4A damit wird in Assembler verglichen, ob eine Eingabe im Bereich A..I liegt
In RAM60.COM : An den Adressen 34CB und 34DF den Wert von vorher 0E in 0F ändern. Resultat: Das blinkende Klick hört auf zu blinken.
Anm.d.HzN: Wieso ? Das Klick blinkt doch garnicht - es bimmelt!
6. Vor den Programmen PACK, RESTORE, CLEANDIR, SAP wird gewarnt, sie haben nämlich die unangenehme Art, die Zeiteinträge zu verschlusen

Das wars was ich in letzter Zeit geschafft habe.

Gruß aus BS
Holger

Anm.d.HzN: Zwei weitere Punkte von Holger habe ich in eigene Artikel gesteckt!

L e s e r b r i e f: Ulrich Taschke, 7500

Hallo!

Nun bin ich schon seit etwa Mai in Eurem Club und habe noch immer keinen "Einstandsbeitrag" im Info geliefert!

Ich weiß nicht, ob es stimmt, aber ich stelle mir vor, daß der MTX-User-Club nur noch selten einfach so Zuwachs bekommt, denn wer kauft sich heute noch einen CP/M-Rechner? Ich habe meinen MTX Ende '84 bei Vobis gekauft, um meine Diplomarbeit drauf zu machen. Und erst jetzt habe ich mich entschlossen, beizutreten.

Und nun sitze ich hier an meinem Rechnerle, mit dem gerade neu erworbenen RAM 6.0. Eigentlich bin ich ja hellauf begeistert, aber bei manchen Dingen stehe ich noch so ziemlich hilflos da. Man bedenke: Für mich gab es bis vor 3 Monaten nur CP/M 2.2 mit PIP, STAT usw. Und jetzt habe ich RAM 6.0, P2Dos und ZCPR! Ein Super-System, aber wenn ich mal schnell eine Datei ansehen wollte, habe ich immer TYPE gehabt... Das DIR gibt es zwar noch, aber wenn das neue System schon Zeiteinträge beherrscht, will ich sie auch sehen! Möglichst mit Dateilängen. Ich habe zwar ACOPIY gern übernommen, trotzdem ist das PIP noch nicht entbehrlich geworden, wenn ich was auf den Drucker schicken will oder an den Bildschirm (TYPE ist ja weg). Der langen Rede kurzer Sinn:

Welche Utilities gibt es überhaupt, was braucht man unbedingt, was ist sinnvoll, was ist überholt, was ersetzt alte Utilities (z.B. Type, PIP)?

In der langen PD-Liste sind reihenweise welche da, aber (wahrscheinlich) gleich mehrere für dieselbe Anwendung und dann noch in allen möglichen Versionen! Für jemand der so plötzlich von 0 auf 100 (sprich von 2.2 auf 6.0) kommt, ist es schwer, sich da zurechtzufinden! Zumal ich kein Programm-Hamster bin. Ich bin bestrebt, möglichst wenige Utilities zu benutzen. So kam ich bisher mit WASH (einem Programm ähnlich wie SWEEP, aber besser), ZAP und den Standard-CP/M-Programmen aus. Die werde ich jetzt wohl nicht mehr brauchen können, da sie die Zeiteinträge und die DU: Angaben nicht können. Und was jetzt?

- Super-Dir, oder was besseres?
- Super-ZAP, DU oder DiJey? *Arm.d.HZN: DiJey*
- welche der vielen Disk-Verwaltungen? Jetzt sind die User-Bereiche ja erst interessant geworden.
- Gibt's was WASH-ähnliches? (NSWEEP etwa?) *Arm.d.HZN: NSWEEP gibt's*

Was ist sinnvoll? Ihr habt doch in Ruhe bisher die besten Programme aussuchen und ausprobieren können. Da müßt Ihr doch eine Menge Erfahrung haben, was womit in welcher Zusammenstellung am besten zu machen ist. Ich möchte nun aber nicht alle 35 bisherigen Infos lesen müssen, um das nachzuholen. Laßt doch mich (und evtl. andere) Neueinsteiger davon profitieren!

Und jetzt zum Schluß noch was allgemeines, was mich interessiert:

- Wieviele Mitglieder hat der MTX-Club Deutschland? *Arm.d.HZN: ca. 120*
- Gibt es auch welche in meiner näheren Umgebung? *Arm.d.HZN: Liste anbei*
- Wie lange existiert der Club schon? *Arm.d.HZN: Mitte 1984*

Ich freue mich aufs nächste Info!

L e s e r b r i e f: Hartmut Traber, 5270

Hartmut Traber

Hohbeulstr. 8, 03.06.90
5270 Gummersbach
Tel.: 02261/65399

Lieber Herbert,

1. Auf dieser Scheibe ist mein System und RAM6, installiert für c't CMOS 512K, 6MHz usw., bootfähig.
Zur Erinnerung: die Hardware-Uhr (F) läßt sich nicht auf volle Anzeige installieren, Spalte 79 ist in allen Bildschirm-Formaten Ende, Spalte 80 ist vorhanden.
Die Eingabe 22 in INST6 für die Anzeigelänge wird bei mir verweigert.

Anm.d.HzN: Wenn Du für 'Von Zeichen' 0 eingibst, dann klappt's

Die Direkteingabe von ESC-Sequenzen scheint, abweichend von RAM43(45) nicht mehr möglich zu sein.

Anm.d.HzN: Doch, aber nur mit vorangestelltem ?. Das liegt am ZCPR.

2. Ganz grundsätzlich bin ich (noch?) kein Freund von Libraries, vielleicht weil ich mich an die zugehörigen Hilfsprogramme noch nicht gewöhnt habe. Die Vorteile z. B. auf einer SRAM-Boot-Disc sehe ich schon, dort stelle ich mir demnächst aber eine eigene *.LBR selbst nach meinen Bedürfnissen zusammen. Diese enthält dann z. B. alles, was ich beim Booten in den Heap bringen will. Auf -Club.???-Disketten sollte in Zukunft aber nicht alles nur noch gesqeezed, gecruncht oder gesqeeecruncht sein. Der Zeitaufwand, um alles zu sichten steht in keinem Verhältnis zu den eingesparten Diskettenkosten.
Ich finde, daß die Übersicht leidet, kann aber durchaus sein, daß diese meine Ansicht falsch ist.

Anm.d.HzN: Zu Libraries habe ich einen Artikel geschrieben, der in diesem Info steht.

3. Ich stelle mal zur Diskussion:
Standard-Format für Club-Discs:
03 sollte bleiben für Nicht-RAM6-Nutzer (die sind selbst schuld!), kann aber auch wegfallen, denn es ist jedem zuzumuten, PD-Nutzer (Michael Keßler) zu sein.
09 sollte Standard-Format werden, da nicht jeder ein 80-Spur-Lw. hat.
Für alle 80-Spur-Lw.-Besitzer schlage ich als Standard-Format für PD- und Klick-Clubs 1B vor, aber nur, weil Du das Format 1C nicht liebst! Bei mir gibt es als Arbeitsdisketten nur noch 1C!

Anm.d.HzN: Hier mein Vorschlag: 03 und 09 bleiben für PD's die Standardformate. Und zwar 09 nur für solche PD's, die RAM x.x voraussetzen. Es gibt einige MTX-ler, die kein 80-Spur-Laufwerk ihr Eigen nennen.
Für 80-Spur-Disketten hat sich eigentlich 0A als Standard eingebürgert, da es auch mit Billig-Disketten verwendet werden können.

4. Nach Aufruf von KI60.com stürzt mein Rechner ab!
(Danke für die RESET-feste RAM-Disc A:!)

Anm.d.HzN: Wie schaffst Du das ??

Mit freundlichen Grüßen
Hartmut Traber

L e s e r b r i e f: Hartmut Traber, 5270

Hartmut Traber

Hohbeulstr. 8, 11.06.90
5270 Gummersbach
Tel.: 02261/65399

Moin, moin Herbert,

- * In die "Schwarze Liste" muß noch SIZE.COM aufgenommen werden!
(Rufe es mal auf: ist ganz interessant, was da gemeldet wird.)
Anm.d.HzN: Ja!
- * Mir kommen am Rechner sitzend immer neue Ideen, die Anderen u. U. Arbeit machen (sollen?):
Wenn ich das Laufwerk z. B. D3: eingeloggt habe und auf z. B. G0 wechseln möchte, gebe ich ein: G:. Ich lande aber in G3(:).
Logisch, aber etwas inconvenient.
Kann man da dem ZCPR 3.3 noch etwas beibringen? Er sollte ohne Angabe eines Users immer nach X0: wechseln!
Anm.d.HzN: Ich hab's mit Aliasen versucht. Klappt leider nicht.
- * Nach langem Grübeln, Nachlesen im RAM6-Handbuch und Konkretisieren meiner Bedürfnisse ist dann noch folgendes Ergebnis 'rausgekommen:
 1. Ich will normal von der c't-SRAM-512-K booten, und zwar ohne je eine Taste zu betätigen. (Das von mir in der Installation gewählte PROMT soll sofort erscheinen, deswegen arbeite ich auch nicht mit Menu).
 2. Auf das Booten von Scheiben in dichten Formaten kann ich verzichten, wenn notfalls auch mit Tastenbetätigung das Booten von Normal-Clubformaten möglich ist (z. B. bei Ausfall der statischen RAM-Floppy, aber die ist bei mir jetzt außerordentlich betriebssicher).
 3. Das bedeutet vermutlich, daß es sich nur noch um eine Überschneidung von EB00-EB20 handelt, wenn man ein 60-K-System mit dem Feature "Booten von der SRAM-Floppy" haben will.
Diese 20 Bytes sollten doch noch zu holen sein ?
Das Handbuch ist außerordentlich gut, professionell gut, sogar noch besser !
Aber wie Du siehst, Einzelfragen kommen doch noch auf.
Da Du alle Software von Michael hast, ich ggf. noch was beisteuern könnte, wäre ein Hardware-Angebot:
Modifizierung des Boot-Eprom's: DM xx.xx
zumindest für mich interessant.
(M. Keßler hat wohl alle Aktivitäten aufgegeben).
Anm.d.HzN: Ich biete gerne die älteren BOOT-Eproms à la Bernd Preusing an.
- * Nachdem ich mal so einiges, eigentlich fast alles bis -CLUB.802 und -CLUB.901 dekomprimiert habe und mit anderen Quellen verglichen habe, Du weißt, daß ich ZCPR 3.3 auch für den Schneider habe, gucke ich etwas durch. Ihr habt Euch ja eine ungeheure Arbeit gemacht! Nicht zu bezahlen, jedenfalls nicht mit DM 60,-- (!). Der Rest steht mir noch bevor.
- * Aber ein weiterer Wunsch kommt da auf:
Load60 -Klixprg.lbr *.klix wäre für mich sehr schön, zumal es auch aus unterschiedlichen Libraries laden kann, die ich mir für bestimmte Anwendungsfälle zusammenstellen und benennen könnte, die ganze Startup-Chose wäre etwas einfacher. XRUN, LX, XRUN, SALIAS können das wohl nicht? (Probiert habe ich es ohne Erfolg).

Hat mich beeindruckt: "Any ideas? YES".

Mit freundlichen Grüßen
Hartmut Traber

V A T I C A L: Eine revolutionäre Programmiersprache?**VATICAL - Die Apocalypse der Programmiersprachen Version 2.2**

VATICAL (TM) erschien im Jahre des Herren 1983 als Creation der legendären Softwarehäuser CRUSYWARE & CRADDLESOFT (TM). Trotz Bedenken ansonsten gut informierter Kreise konnte ihr bereits 1984 Serienreife erteilt werden. VATICAL (TM) ist in der nunmehr vorliegenden vorläufig endgültigen Version 2.2 absolut und vorsätzlich inkompatibel zu allen vorhandenen Betriebssystemen, Prozessoren und Expertenmeinungen und hat nicht zuletzt deshalb allerbeste Marktchancen.

Ursprünglich war VATICAL (TM) sehr nahe an Pascal orientiert. In der ersten Entwicklungsphase konnte jedoch erfreulicherweise eine Annäherung an BASIC und FORTRAN erreicht werden, ohne das auf die Unzulänglichkeiten von PASCAL verzichtet werden mußte. Selbst die sattem bekannten Mängel von BASIC und FORTRAN konnten erfolgreich implementiert werden.

VATICAL (TM) ist weder über Compiler noch Interpreter lauffähig, sondern arbeitet über einen Absoluter, der im Gegensatz zu den schäbigen FORTH Versionen keinen Stapelspeicher mehr benötigt, sondern eine 'EWIGE LISTE'. Diese 'EWIGE LISTE' kann weder überschrieben noch gelöscht werden. Allenfalls der Befehl 'DOOMSDAY' erstellt eine Liste der erfolgten Eingaben und läßt VATICAL (TM) entsprechende Entscheidungen treffen. Sie decken sich nicht unbedingt mit Absicht und Willen des Benutzers.

VATICAL (TM) besitzt verschiedene Schnittstellen. So wird das Druckerinterface über 'PRAY' angesprochen. VATICAL (TM) überprüft alle Ausgaben an den Drucker und behält sich leichte Korrekturen und Kommentare vor. Zur nicht geringen Überraschung der Erfinder verfügt sie auch über ein Disketteninterface und betreibt es über die Befehle 'MISSION' und 'MERCY'. Allerdings können ausschließlich indexgeprüfte HolyAccess Dateien bearbeitet werden. Der Default Index von VATICAL (TM) vermerkt Änderungen an ihm in der 'EWIGEN LISTE'. VATICAL (TM) unterstützt den Betrieb von maximal drei Laufwerken, die bindend die Bezeichnung 'FATHER', 'SON' und 'SAINT' haben. Dateien und andere Files werden grundsätzlich über das Laufwerk 'SAINT' bearbeitet. Das Laufwerk 'SON' bietet Hilfs- und Utilityfunktionen, während das Laufwerk 'FATHER' ausschließlich das Betriebssystem verwaltet.

Im Gegensatz zu den herkömmlichen und zumeist veralteten Disk-Betriebssystemen benötigt VATICAL (TM) weder ein Inhaltsverzeichnis noch Sektoren oder gar Spuren. Der Diskettenplatz wird in sieben Büchern aufgeteilt und jedes Buch wieder in Kapitel. Die jeweiligen Kapitelinhalte verwaltet das Laufwerk 'FATHER' in der Datei 'APOCALYPSE'. Diese Datei gehört zum Lieferumfang und erlaubt den Besitzer keinerlei Änderungen. Zu dieser Datei nicht konforme Aufzeichnungen werden grundsätzlich nicht am Bestimmungsort abgelegt, sondern landen allenfalls in der 'EWIGEN LISTE'.

VATICAL (TM) erlaubt die Befehlseingabe sowohl in lateinischer als auch englischer Sprache. Im Manual wird jeweils der bekanntere begriff benutzt. Für eventuell notwendige Übersetzungshilfen kann jedes Lexikon herangezogen werden.

Nach dem Einschalten des Rechners meldet sich VATICAL (TM) mit dem systemtypischen Cursor '+' und erwartet den Aufruf oder die Eingabe eines Programmes.

V A T I C A L: Eine revolutionäre Programmiersprache?

Programme müssen grundsätzlich mit 'LITURGY:' beginnen und mit 'AMEN' enden. Im Programm selbst darf nach Herzenslust umhergesprungen werden. Es darf auch aus dem Programm herausgesprungen werden, was auch immer dies bedeuten oder bewirken mag. So kennt VATICAL (TM) die ungewöhnlichen Befehle wie:

GOLEFT, GORIGHT, GOSUB, GOUP, GODOWN, GOAWAY, GOHOME,
GOASCENSION, GOHELL, GOGO, GOWEST, GOELSE, GOANYWAY, GONOT
und GOTOGODOT.

Ähnlich den freiprogrammierbaren Sprunganweisungen kann ein beliebiger Teil des Programmes als Unterprogramm erklärt werden, z. B. alles, was mehr als 20 Zeichen weit rechts in einer Zeile steht. Grundsätzlich kann jeder Teil eines Programms einen beliebigen anderen Teil als Unterprogramm definieren. Der Programmteil, der die meisten Unterprogramme findet, ist das Hauptprogramm. Demgemäß existiert in VATICAL (TM) kein Befehl 'RETURN', sondern die Rücksprunganweisungen 'SEARCHME', 'WAY NEXT', 'RESTART', 'DOBEST' und 'TRY AGAIN'.

VATICAL (TM) bietet dem geneigten Benutzer zahlreiche Möglichkeiten zur Schleifenbildung innerhalb der Programme:

FOR-NEXT-DO	FOR-NEXT-DO-NOT
FOR-NEARER-DO	UNTIL-BETTER-USE-DO
UNTIL-NO-ERROR-PERFORM	FOR-WHAT-REASON
WHY-FOR	WHAT-ELSE
WHY-NOT-PLEASE	WHY-DONT-WE-DO-IT
WHAT-SHALLS	TRY-UNTIL
HOPE-LESS	UNTIL-NO-FUTURE-DO
JUMP-ANYWHERE-UNTIL	WAIT-UNTIL-DOOMSDAY
IF-NO-TEXT-THEN-TEXT	WAIT-FOR-GODOT

Glücklicherweise besitzt VATICAL (TM) Zahl- und folgenreiche Meldungen von Syntaxfehlern des Programms, gelegentlichen Systemabstürzen, Dateiüberschreibungen und generell mitgelieferten Fehlern des Betriebssystems.

Grundsätzlich meldet VATICAL (TM) einen auftretenden Fehler wie folgt:

```

MY SCORE   YOUR SCORE   CHAMPION   CHAMPIONS'S SCORE
*** ALL THE BEST ***
**      SO LONG      **

```

Unnötig zu bemerken, daß jeder Fehler unnachsichtig in der 'EWIGEN LISTE' eingetragen wird. Mit einer Ausnahme: Findet VATICAL (TM) besonders elegante und funktionstüchtige Programme (Maximalwert = 3 Fehler pro Zeile), kann der Benutzer mit dem Befehl 'INDULGENCE' Einträge in der 'EWIGEN LISTE' stornieren.

Bei Überlauf des Variablenspeichers betreibt VATICAL (TM) im Gegensatz zu Schundsprachen wie BASIC keine Garbage Collection, sondern wandelt diese Funktion in eine 'VALUE COLLECTION' um. Alle Variableninhalte, die VATICAL (TM) als brauchbar ansieht (z. B. reale, positive Zahlen oder Zeichenketten mit originellem Inhalt und kulturellem Niveau) verlegt das Betriebssystem in die Datei 'RECORDS' auf der 'FATHER' Diskette. Die restlichen Einträge werden in der 'EWIGEN LISTE' abgelegt und der Speicher wird gelöscht. Einträge, die der Benutzer weiterverarbeiten will, müssen vorher in der Datei 'ESTATE' abgelegt werden.

Variablen werden mit dem Betriebssystem vereinbart und an geeigneter Stelle getauft. Erst die namentliche Taufe einer Variablen, z. B. in der Form 'ST.KEVIN CHRISTIANED AS INTEGER', läßt das Betriebssystem die Variable erkennen.

V A T I C A L: Eine revolutionäre Programmiersprache?

Alle Namen von Variablen müssen im Programmkopf vereinbart werden. VATICAL (TM) überprüft Art, Sinn und Zweck dieser Vereinbarungen und behält sich leichte Korrekturen vor. Die Namen der Variablen müssen zwingend mit 'ST.' beginne. Daran angehängt wird der eigentliche Variablenname und zwar:

Integerzahlen	Namen von Heiligen (1500 - 1800)
Reale Zahlen	Namen von Heiligen (1800 - 1900)
Genauere Zahlen	Namen von Heiligen (1900 - heute)
Charakterzeichen	Namen bedeutender Kirchenpersonen
Textketten	Namen von Propheten (Altes Testament)
Einfache Arrays	Namen von Päpsten
Mehrdim. Arrays	Namen von Aposteln (Neues Testament)

Besonderes Augenmerk gilt hier den Charakterzeichen. Sollte eine Variable mit ST.LUTHER oder aber noch schlimmeres bezeichnet werden, hat dies katastrophale Folgen für den Benutzer. Alle Variablennamen sind auf- und abwärts kompatibel, allerdings vermerkt VATICAL (TM) hierbei Verwechslungen mit 'NOT YET' oder 'FLATTERY'.

Die Darstellung von Zahlen im negativen Bereich erfolgt durch den Zusatz von 'TEMPTATION'. Die Schreibweise von

'TEMPTATION OF ST.KEVIN = 10000'

ergibt für die Integervariable ST.KEVIN den Wert '-10000'. Variablenwerte können in VATICAL (TM) grundsätzlich nur addiert werden und zwar in dieser Form:

'ST.KEVIN AND ST.MARTIN PRAISE FOR ST.RESULT'

Das Ergebnis der Addition steht dann logischerweise in 'ST.RESULT'. Die Zuweisung es Wertes 'NULL' zu einer Variablen Bedarf der Form:

'ST.KEVIN IN POVERTY'

Die Darstellung negativer Realwerte erlaubt VATICAL (TM) ausschließlich über die Bezeichner 'JUDAS', 'KAIN' und 'ROCKNROLL'. Alle Variablenbezeichner gelten nicht nur global im Programmbereich, sondern auch noch nach dem Abstellen des Rechners.

VATICAL (TM) beinhaltet einen leistungsfähigen Programmeditor, der über den Befehl 'MEA CULPA' aufgerufen wird. Der Editor unterscheidet zwischen:

§ = einfachen Sünden

\$ = Erbsünden

\$\$ = Todsünden

Sollten im Programm Todsünden auftreten, muß der Benutzer von VATICAL (TM) den Editor mit 'MEA MAXIMA CULPA' aufrufen und anschließend je nach Zahl der Fehler mit x-mal 'ROSARY' eingeben.

In allen Fällen verlangt der Programmeditor die Eingabe des Benutzernamens. Er wird in der 'EWIGEN LISTE' abgelegt. Der Editor vergleicht Sinn, Art, Zweck, Ausführung und Effizienz des Programms und unterstützt demgemäß den Benutzer. Dazu listet er zunächst die fehlerhafte Zeile auf dem Monitor aus. Der Befehl 'MISSION' liefert eine Korrektur, wie sie VATICAL (TM) für ratsam hält. Der Befehl 'DEMISSION' erlaubt eine benutzerspezifische Eingabe. Das Ende des Programmeditors wird durch die Eingabe 'SAKE' abgeschlossen. Zur Online Fehlersuche trägt VATICAL (TM) durch verschiedene Befehle bei.

Die Eingabe 'DEVILINSIDE' löscht das Programm selbst und auch alle ähnlich lautenden Programme.

Der Befehl 'INQUISITION' suspendiert das laufende Programm, setzt selbstständig Fehler ein und startet es neu. Der Ablauf wird dann protokolliert und mit dem ursprünglichen Zustand verglichen. Auftretende Fehler werden mit den Einträgen in der 'EWIGEN LISTE' verglichen und dementsprechend dokumentiert.

V A T I C A L: Eine revolutionäre Programmiersprache?

Der Befehl 'CONFESS' schließlich listet alle augenscheinlichen und wahrscheinlichen Fehler auf und erlaubt über die Eingabe 'AVE MARIA' eine sofortige Korrektur.

Eine weitere Stärke von VATICAL (TM) zeigt sich in der Behandlung von logischen Operationen und der geneigte Benutzer wird viel Freude mit bestimmten Systemeigenheiten erleben.

Die Funktionen 'AND', 'ANDC', 'ADD' und 'ADDC' verlangen zwingend gleichgeschlechtliche Variablen auf beiden Seiten. Die Form

'ST.MARTIN = ST.MICHAEL ADDC ST.HELENA'

erzeugt mindestens eine faustdicke Fehlermeldung.

Als Besonderheit logischer Operationen bietet VATICAL (TM) einige ungewöhnliche Optionen:

'ADDAF'	=	ADD-AND-FORGET
'CAC'	=	CASH-AND-CARRY
'ENDC'	=	END-OF-CARRY
'NENA'	=	NEITHER-NOR-ANYTHING-ELSE

VATICAL (TM) beinhaltet mehrere Anwenderprogramme, deren intensive Benutzerfreundlichkeit und ungewöhnlich bedienerorientierte Menüstruktur für sich sprechen.

1. Das Systemprogramm VATICAL (TM) selbst
2. VATICALC (TM) - ein spezielles tabellenorientiertes Rechenprogramm zur Verarbeitung von Kollektivspenden. Sowohl Spendenbetrag, als auch Name, Adresse, Beruf, Nettoeinkommen und streng vertrauliche Angelegenheiten des Spenders werden in der 'EWIGEN LISTE' vermerkt. Auf der Basis der Jahreskollektensummen können Prognosen über zu erwartende Einnahmen getroffen werden. Ein spezieller Programmteil erlaubt die finanzielle Berechnung von Kirchenrenovierungen und den Vergleich mit den erwartbaren Kollektenbeiträgen.
3. BIBLIOSTAR (TM) - ein Textverarbeitungsprogramm, das Synonym zu SPELLSTAR (TM) bereits alle relevanten Bibeltexte enthält und die Ausarbeitung einer 20 Minutenpredigt in weniger als 2 Minuten erlaubt. Eine automatische Übersetzung Latein/Deutsch ist in dieser Option bereits enthalten.
4. MULTICONFESS (TM) - eine Datenbank zur schnellen Speicherung und Bearbeitung von Daten, die während Beichten und vertraulichen Gesprächen auftauchen. Das Programm speichert die Daten in der 'EWIGEN LISTE'.
5. PEACEMAN (TM) - ein Spielprogramm, in dem angreifende Kommunisten und Atheisten durch die schnelle Eingabe von Bibelzitate abgewehrt werden müssen.
6. INRI-1-2-3 (TM) - ein äußerst anwenderorientiertes Programm zur schnellen Berechnung von Abläßen, zu betenden Rosenkränzen und Zahl, Durchmesser und Länge von Sühnekerzen. Beinhaltet eine umfangreiche und sich selbst erweiternde Datei von gängigen Sünden und Lastern. Ein Muß für die datenverarbeitende Beichte. Alternativ kann auch die Zahl der zu erwartenden Jahre im Fegefeuer gelistet oder auch - für das Beichtkind - über einen Drucker ausgegeben werden.
7. HOLYSOFT (TM) - ein Expertensystem für den kirchlichen und geistlich orientierten Anwender. Basiert auf Frage- und Antwortsystem und liefert in kürzester Zeit und ca. 3 Detailrecherchen schlagkräftige und schlüssige Argumente, denen sich auch der hartgesottenste Ignorant und Klerikalspötter nicht verschließen kann.
8. TRUEBASE III (TM) - eine selektive Datenbank, bei der die Ein- und Ausgabe streng konform zur 'EWIGEN LISTE' läuft. TRUEBASE-III packt die Dateien auf kleinstes Format zusammen. Nur wirklich speicherwürdiges wird gespeichert.

V A T I C A L: Eine revolutionäre Programmiersprache?

9. SYSZINK (TM) - ein Programm, das fast unbeschränkten Zugriff und Editierungshilfen auf die 'EWIGE LISTE' bietet. Voll Menuegesteuert breitet es die bisherigen Fehlermeldungen und Fehlgriffe vor dem Benutzer aus.

VATICAL (TM) erlaubt das Verbinden (=Linken) mehrerer Anwenderprogramme. Maximal 4 Programme können gelinkt werden. Müssen allerdings unter einem festen Programmnamen abgelegt werden. Der 'LINKER' wird durch den Befehl 'HIGHMASS' aufgerufen und verlangt in vorgegebener Reihenfolge nach den Programmen:

'KYRIE'	(Variablenvereinbarungen, Arraydimensionierung, Konstantendefinition, Dies & Das)
'GLORIA'	(Funktionen und Prozeduren)
'CREDO'	(Programm 1)
'SANCTUS'	(Programm 2)
'BENEDICTUS'	(Programm 3)
'AGNUS DEI'	(Programm 4)

Der Linker wird durch den Befehl 'IN NOMINI:' aufgerufen und erwartet zuerst die Identifizierung durch den Benutzer. Dann erfolgt die Überprüfung auf das Vorhandensein der einzelnen Programme, Speicherbedarf und richtige Syntax.

Diese Prozedur benötigt überraschend viel Zeit. Der geneigte Benutzer wird daher während dieser Zeit mit saisonal passenden Zitaten aus kulturell hochstehender Literatur unterhalten. Es wird ausdrücklich gewarnt, während dieser Zeit das Terminal zu verlassen, da VATICAL (TM) gelegentlich und nach peinlich geheimgehaltenem Zufallsmustern Verständnisfragen an den Benutzer stellt und die Zeit bis zur Beantwortung speichert. Zuweilen reagiert VATICAL (TM) unter bestimmten Umständen sehr übel, falls der Benutzer nicht available sein sollte.

Allgemein können für den Umgang mit VATICAL (TM) die folgenden 10 Regeln besonders hervorgehoben werden:

1. neben VATICAL (TM) sollten keine anderen Betriebssysteme im Rechner laufen
2. Von VATICAL (TM) darf keine Raubkopie erstellt werden
3. An Sonn- und Feiertagen darf VATICAL (TM) nicht benutzt werden
4. Die Hersteller von VATICAL (TM) versprechen seine Lauffähigkeit. Daran sollte auf keinen Fall gezweifelt werden.
5. Dateien und Programme sollten weder überschrieben noch gelöscht werden. Dies besorgt zu gegebener Zeit das Betriebssystem selber.
6. Aus anderen Betriebssystemen dürfen weder Dateien noch Programme kopiert werden.
7. Aus VATICAL (TM) dürfen weder Programme noch Dateien auf andere Betriebssysteme kopiert werden.
8. In Dateien dürfen nicht widersprüchliche Daten geschrieben werden.
9. Andere Betriebssysteme sind auch nicht besser.
10. Andere Programme sind immer schlechter.

Friede sei mit Euch.
Und mit eurem Betriebssystem.
Bis zum nächsten Absturz.
Amen.

New Word: Grafik und Sonderzeichen

NEWORD UND GRAPHIK

=====

Im Info 13 schrieb Bernd Preusing einen sehr guten und interessanten Artikel: 'NewWord: Printer-Overlay'. Gleichzeitig war ein Public Domain (CLUB.009) fertig mit verschiedenen Druckertreiber, u.a. NWPDMX80. Mit diesem Treiber konnte man schon mit unserem guten alten DMX80 einen Rahmen auf Papier drucken via das Kommando ^PP.

In seinem Artikel erwähnte Bernd auch das Folgende: 'Mit ^PP habe ich noch Großes vor, also vorl. nicht benutzen!' Dieses Große war vielleicht einen ganz neuen Treiber mit mehr Graphik, oder? Aber,... Bernd hat schon lange Zeit unsere Club verlassen und 'Meinen Wunsch-Treiber für der FX-80 zu schreiben, wird wohl noch etwas dauern'.....

Als Gymnasiallehrer Chemie und Physik benutze ich häufig griechische Buchstaben, wie alfa, beita, rho, omega, u.s.w.. In meine Texte verarbeite ich auch viele Rahmen für Tabellen mit Daten aus Labo-Experimenten. Ich brauche dringend einen neuen Treiber für meine Lektionen. Also habe ich mich an die Arbeit gesetzt und der Treiber von Bernd angepaßt und erweitert mit ca. 90 graphische Zeichen. Als Beispiel diene der IBM-Graphik für die Zeichen mit Kode 46 bis 127. Die Zeichen (33 bis 45) werden benutzt um die fehlende griechische Buchstaben zu definieren.

Das Resultat sieht so aus:

NEWORD UND GRAPHIK...

<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="padding: 2px;">τ ω Γ Δ</td><td style="padding: 2px;"></td></tr> <tr><td style="padding: 2px;">Σ φ Ω</td><td style="padding: 2px;"></td></tr> </table>	τ ω Γ Δ		Σ φ Ω		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 2px;">α β γ δ ε μ</td> <td style="padding: 2px;">φ η θ λ ν π ρ σ</td> </tr> <tr> <td style="padding: 2px;">ω</td> <td style="padding: 2px;">e.d.</td> </tr> </table>	α β γ δ ε μ	φ η θ λ ν π ρ σ	ω	e.d.	<div style="text-align: right; margin-bottom: 10px;">■</div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;"> $\frac{1}{4} + \frac{1}{4} = \frac{1}{2}$ $\sqrt{81} = 9$ </td> <td style="padding: 5px;"> $A \approx B$ </td> </tr> <tr> <td style="padding: 5px;"> </td> <td style="padding: 5px;"> </td> </tr> <tr> <td style="padding: 5px;"></td> <td style="padding: 5px;"> $10 \gg 3$ $3 \ll 9$ </td> </tr> </table>	$\frac{1}{4} + \frac{1}{4} = \frac{1}{2}$ $\sqrt{81} = 9$	$A \approx B$				$10 \gg 3$ $3 \ll 9$			
τ ω Γ Δ																			
Σ φ Ω																			
α β γ δ ε μ	φ η θ λ ν π ρ σ																		
ω	e.d.																		
$\frac{1}{4} + \frac{1}{4} = \frac{1}{2}$ $\sqrt{81} = 9$	$A \approx B$																		
	$10 \gg 3$ $3 \ll 9$																		
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;"> $F = \frac{1}{4 \cdot \pi \cdot \epsilon} \cdot \frac{Q_1 \cdot Q_2}{r^2}$ </td> <td style="padding: 5px;"> <div style="border: 2px solid black; width: 50px; height: 20px; margin: 0 auto;"></div> </td> </tr> <tr> <td colspan="2" style="padding: 5px; text-align: center;"> NEWORD UND GRAPHIK </td> </tr> <tr> <td colspan="2" style="padding: 5px;"> $\sin(\alpha + \beta) = 2 \sin \alpha.$ </td> </tr> <tr> <td colspan="2" style="padding: 5px;"> $\Delta s \sim \Delta t$ </td> </tr> <tr> <td colspan="2" style="padding: 5px;"> $\div \infty \emptyset \hat{=} \equiv \pm \leq \leq$ </td> </tr> </table>	$F = \frac{1}{4 \cdot \pi \cdot \epsilon} \cdot \frac{Q_1 \cdot Q_2}{r^2}$	<div style="border: 2px solid black; width: 50px; height: 20px; margin: 0 auto;"></div>	NEWORD UND GRAPHIK		$\sin(\alpha + \beta) = 2 \sin \alpha.$		$\Delta s \sim \Delta t$		$\div \infty \emptyset \hat{=} \equiv \pm \leq \leq$		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="padding: 5px; text-align: center;"> MTX User-Club Deutschland </td> </tr> <tr> <td style="padding: 5px;"> $\sin \beta$ </td> <td style="padding: 5px;"> $A \longleftrightarrow B$ </td> </tr> <tr> <td colspan="2" style="padding: 5px;"> </td> </tr> <tr> <td style="padding: 5px;"> $\int 2x \cdot dx = x^2$ </td> <td style="padding: 5px;"> $A + B \longrightarrow C$ </td> </tr> </table>	MTX User-Club Deutschland		$\sin \beta$	$A \longleftrightarrow B$			$\int 2x \cdot dx = x^2$	$A + B \longrightarrow C$
$F = \frac{1}{4 \cdot \pi \cdot \epsilon} \cdot \frac{Q_1 \cdot Q_2}{r^2}$	<div style="border: 2px solid black; width: 50px; height: 20px; margin: 0 auto;"></div>																		
NEWORD UND GRAPHIK																			
$\sin(\alpha + \beta) = 2 \sin \alpha.$																			
$\Delta s \sim \Delta t$																			
$\div \infty \emptyset \hat{=} \equiv \pm \leq \leq$																			
MTX User-Club Deutschland																			
$\sin \beta$	$A \longleftrightarrow B$																		
$\int 2x \cdot dx = x^2$	$A + B \longrightarrow C$																		

Schön nicht? Aber, das Schönste kommt noch!

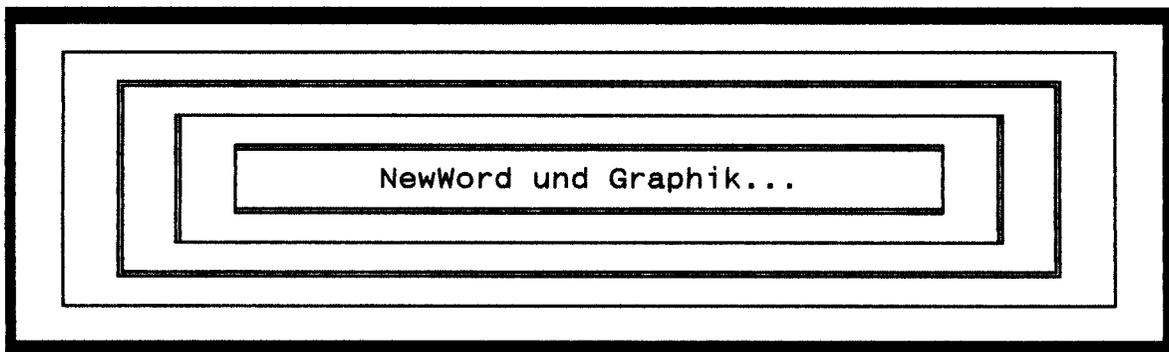
New Word: Grafik und Sonderzeichen

Da gibt doch noch ein kleines Problem: das Resultat kann man nur anschauen wenn der Text gedruckt ist. Meiner Wunsch war nun: dieselbe graphische Zeichen auf das Schirm zu bringen! Ich meinte daß die Hercules-Karte von Horst Kupka die Lösung war, aber nein: mit dieser Karte kann man z.B. umschalten zwischen vier verschiedene Zeichensätze, aber keiner Text zusammen mit Graphik!!! Die Lösung war dennoch sehr einfach und zu benutzen durch ALLE FDX-Besitzers (Hercules-Karte ist gar nicht nötig). Auf der 80-Zeichenkarte sitzt ein ROM (IC A9) oder EPROM mit Graphik-Blöckchen. Diese Zeichen sind, soviel ich weiß, noch niemals benutzt in einem Programm. Dieses ROM wird nun ersetzt durch ein 2732 EPROM mit einem graphischem Zeichensatz.

In NewWord (nach patchen mit Herberts NWPATCH) mit der neuen Bildschirmtreiber steht an der Adresse #03B7 die Werte #20 (background). Diese Werte muß ersetzt werden durch #80 (Graphik) mit z.B. MONI oder ähnlich. Beim Editieren vom Text mit NewWord erscheinen die Graphik-Zeichen auf das Schirm mit dem Toggle-Kommando ^PX.

Einfach nicht? Das Kommando ^PX wird jedoch durch NewWord angewendet für 'Strike-out'-Text. Kein Not, mit dem neuen Treiber kann man nun 'Strike-out'-Text drucken mit dem Kommando ^PK.

Für Leute mit einem Epson LQ-Drucker habe ich einen anderen Treiber angepaßt, der die eingebaute IBM-Zeichen benutzt. Das Resultat mit einem Epson LQ-Drucker (24 pins) ist selbstverständlich viel schöner:



Und nun die praktische Seite.

Da verschiedene NewWord-Versionen im Umlauf sind und ich keine Lust habe ein Patch-Programm zu schreiben für jede Version schlag ich das Folgende vor: send mir eine Diskette (Format 03!) mit System und deiner NewWord-Version (komplett: NW.COM, NW.OVR und NWMSG.S.OVR) zusammen mit der ROM (IC A9 von 80Z-Karte) wenn dieses IC ein EPROM 2732 ist! Wenn dieses IC kein EPROM ist, dann kann ich ein 2732-EPROM mitliefern für den Preis van DM 10,- Für das Alles (Patchen von NW.COM und NWMSG.S.OVR, Treiber für DMX80 oder LQ-EPSON, wischen und brennen von DEINE EPROM, einschließlich die Kosten für Porto und Verpackung) zahlst Du nur DM 10.- (Bankschein einschliessen bitte).

Wenn Du keine 2732 EPROM mitschickt zahlst Du DM 10.- extra, also ingesamt DM 20.- (Bankschein einschliessen bitte).

Erik D'Hondt Wilgstraat 25A B-9440 EREMBODEGEM Belgien