

MTX *User-Club Deutschland*

Info 33
10.08.1989

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.- (75 Seiten) 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
A s s e m b l e r	
Cursortasten-Abfrage	Seite 3
T u r b o - P a s c a l	
Cardinals	Seite 5
Zeileneditor	Seite 5
C P / M	
ZCPR2	Seite 6
MENU	Seite 10
CAT	Seite 11
ZEX & NewWord	Seite 14
SUB - ZEX - MENU	Seite 16
L e s e r b r i e f	
Hartmut Traber, 5270	Seite 17
S o f t w a r e	
Wo's was zu kaufen gibt	Seite 18
KLICK.007	Seite 23
COMPARE	Seite 24
BACKitUP	Seite 26
Neues in der PD-Welt	Seite 27
H a r d w a r e	
Reparatur-Tip	Seite 19
Tastatur-Zeichensatz ?	Seite 21
Port-Adressen	Seite 22
FDX-OMTI-Adapter	Seite 23
R A M 5.x	
Planung	Seite 28

Preis für dieses Info: DM 9,80

Redaktionsschluß für's nächste Info: 09.09.89

Angebotsliste

(Herbert zur Nedden, 2000)

Da die Post das Porto schmerzlich erhöht hat, und die Angebotsliste nun achtseitig wird, wird sie leider auch etwas teurer: DM 1.70 statt DM 1.50. Die achte Seite ist übrigens diese Seite mit dem Inhaltsverzeichnis des Infos.

WICHTIGES:

(Herbert zur Nedden, 2000)

Es ist schon erstaunlich, daß viele Club-Mitglieder, die ein Info erhalten, auf dessen Umschlag steht, daß der Kontostand dringend einer Auffrischung bedarf, d.h. unter DM 12.- gesunken ist, das Konto erst mehrere Wochen später auffüllen. Also, ich hätte das bis dahin längst vergessen. Vorschlag zur Güte: gleich auffüllen! Ich habe nämlich denen, die das Info 32 mangels Guthaben nicht erhalten haben, einen entsprechenden Hinweis geschickt, und so vierzehn Mitglieder 'aus der Versenkung geholt'.

Liebe Leserin, lieber Leser,

es ist kaum zu glauben, wie dünn dieses Info ist, nicht wahr ? Es hilft halt nichts, mehr hatte ich einfach nicht.

Dafür hat sich auf dem PD-Sektor einiges getan. U.a. ist die James-Bond-KLICK mehr als voll, d.h. ohne Claudio Romanazzis Festplattentreiber haben wir über 500 kiloByte (oder soll ich sagen ein halbes Mega ?) zusammen, so daß KLICK.008 schon angefangen ist. Klar, DiJey ist auch dabei - und der hat viel dazugelernt, was Grund genug für ein 54 kB-Doc liefert. Mit DiJey weist Olaf auch gleich nach, daß ein KLICK-Overlay (hier DiJey) ein Anzeige-.KLX (hier DJ-Help) aufrufen kann, so selbiges geladen ist: .KLX calls .KLX. Auch der Cache - den Manfred Flume auf eigenes Risiko testet und nicht mehr weiß, wie je er daohne arbeiten konnte - ist mit von der Partie. Michael Keßler hat FORMSTAR und alles was damit artverwand ist beigesteuert, sowie die Ausgabe einiger Programme aus dem RAM 4.x-Umfeld optisch verändert. Im CP/M-PD-Bereich finde ich den Dateivergleicher COMPARE von der MT-Pascal Users-Group, der auf CLUB.045 landen wird, recht pfiffig. Verblüfft hat mich das Directory-Programmchen von SIG/M.060: es ist lernfähig: Ich kann dem Teil klar machen, welche Dateien ich bei der Anzeige des Direktories nicht sehen will - d.h. so'n Gemülle, welches ich auf jeder Scheibe tendenziell drauf habe wie z.B. XDIR.COM oder COPYD.COM zeigt mir dieses Programm nimmer an! Außerdem zeigt dieses Programm auf, wie man selbstmodifizierende CP/M-Programme erstellt! Helmut Grothe hat einen Plottertreiber erstellt und auf PD gegeben der nicht nur einfach Stifte wechselt und Figuren zeichnet, sondern auch noch Macros unterstützt.

Kürzlich las ich übrigens eine Bemerkung über die Architektur der 8086: 'Den Erfindern dieser CPU ist es ja bekanntlich gelungen, 32-Bit-Adressen so zu gestalten, daß man damit 1 MByte adressieren kann, und ohne Aufwand sogar ganze 64 kByte'.

Elektroniker der CP/M-UG HH, d.h. der Zeh-Peh-Emm Jusä-Kruhpf Hamburg, verriet mir, daß die TTL-IC's von SGS häufig nur Ärger bereiten würden, während die von Texas (zumindest die erster Wahl (HW-Elektronik)) nicht!

```
Besteht Interesse an einer Sammelbestellung ? <-----+--+
CASE Antwort of                                     !!
  'Nein','Nö','Nee','No', .....: Goto Nächster Punkt      !!
  'Ja','Mmm','Yes',.....:      Lies Weiter                 !!
  'Weiß nicht',.....:         Frage 'Was weißt Du ? '; Goto + !
  else:                         Goto -----+-----+
ENDCASE
```

Was soll die Sammelbestellung umfassen ? Hast Du evtl. Preise für Dinge, die von Interesse sein könnten ? Wenn Ja, bitte Preise + Quelle an mich.

Unlängst versuchte ich einem DoMessTiquen (= MSDOS-User) klar zu machen, daß unsere Tastatur im Interrupt abgefragt wird, die Maus hingegen über die RS232 sich selbst per Interrupt meldet. 'Na und, bei mir wird jeder DOS-Aufruf per Interrupt vorgenommen' war die Antwort. Tja, es gibt dank Intel (dem Hersteller der 'MSDOS-CPU') leicht Mißverständnisse zwischen Intellern (≠ Intellektuellen, sondern = Intel-CPU-Usern) und denen, die die Z80/Z280- oder 68000er-Welt kennen. Wir (d.h. letztere) nennen Interrupt die von außen per Signal an einem eigens dafür vorgesehenen CPU-Pin ausgelöste Arbeitsunterbrechung, die (fast) sofort, egal, was die CPU gerade tut, durch Aufruf einer Interrupt-Routine (wie z.B. unser Tastatortreiber), beantwortet wird. Dann gibt es noch Traps (= Fallen), die wie Interrupts funktionieren, jedoch per Software ausgelöst werden, die Z80-RST-Befehle, die Calls auf die Adressen 0h, 8h, ..., 38h als Ein-Byte-Befehle bereitstellen sowie System-Calls, d.h. Calls, die es bei CPUs, die getrennte User- und System-Speicherbereiche (wie die Z280) unterstützen, ermöglichen aus dem User-Bereich einen System-Einsprung aufzurufen. Intel hingegen bezeichnet so ziehmlich all das oben genannte als Interrupt - die DoMessDos DOS-Calls (DoMessTiquen spechen von DOS-Interrupts) sind soetwas wie unsere RST-Befehle oder Traps, also eigentlich nichts als besondere aber doch lumpige Calls! Kein Wunder, daß unter MSDOS alles per Interrupt läuft, oder ?

*An Hardy gilt End
Euer Herbert zur Nedden*

C l u b: Korrektur & Nachtrag / Fragen & Antworten**Korrektur und Nachtrag**Uhren-Port 3Eh: (Herbert zur Nedden, 2000)

Gerhard Witzel bietet einen FDX-ECB-Adapter und einen FDX-OMTI-Adapter (s.u.) mit Platz für eine Hardware-Uhr an, bei der die Uhr statt auf Port-Adresse 7 auf der Hauptplatine auf Port-Adresse 3EH kommt. Das ist ja gut und schön, da dadurch Port 7 wieder frei wird - nur daß RAM43 und DATE4 auf Port 7 zugreifen. Damit Ihr nicht Disassemblieren müßt, hier die Patch-Informationen, bei denen jeweils ein 'LD C,7' in einen 'LD C,3EH' geändert wird:

Datei:	DATE4.COM	RAM43FDX.COM	RAM43SDX.COM
Adresse:	0876h	4FFEh	4FE8h

MOUSE.KLX: (Herbert zur Nedden, 2000)

Evtl. war meine Anleitung nicht deutlich genug: Wenn Du meinen Maustreiber einsetzt, darfst Du RSINIT nicht verwenden - zumindest nicht für die betroffenen RS232 bzw. den DART.

Außerdem wird der CTC von dem fertigen MOUSE.KLX nur dann auf 1200 Baud initialisiert, wenn der 74LS193, der den Zähl-Takt für den CTC liefert, mit 4 MHz versorgt wird. D.h. wer mit 4 oder 8 MHz arbeitet hat kein Problem. Die 5- und 6-MHz-Maschinen sind da unpraktischer. An Adresse 151h steht die Time-Constant für den CTC, die für 4 MHz, 1200 Baud: 10h = 16 ist.

Vermutung: Time-Constant für 1200 Baud = 4 mal CPU-Frequenz in MHz.

Hard-Disk-Port: (Herbert zur Nedden, 2000)

Für den Festplatten-Controller (=HardDisk-Controller) sind zwei Port-Adressen vorgesehen, die auch unter RAM 5.x bleiben:

- auf dem ECB-Bus: Basis E8h
- auf dem FDX-Bus: Basis 48h

Je nach HardDisk-Controller belegt dieser vier (z.B. OMTI) oder acht (z.B. c't) Port-Adressen. Die Basis-Adresse bleibt allerdings eine der o.g.

CFIG5.KLX (Herbert zur Nedden, 2000)

Michael Keßler hat zwar meine Formaterkennung technisch verfeinert, allerdings ist mein CFIG43 ein einem Punkt besser: Es läuft auch auf einer SDX!!! Michaels Programme, die automatisch das Format erkennen sind nur auf der FDX lauffähig - getreu dem Motto "Ich hab's für mich geschrieben, wer möchte soll's auch nutzen" - außerdem kann er den SDX-Modus nicht testen.

Fragen und Antworten

F: (Andreas Fischer, CH-4303)

Wie kann man die Cursor-Tasten bzw. einen Joystick in ein Turbo-Programm einbinden? Wo sind die entsprechenden Adressen, die mit diesen Tasten angesprochen werden?

A: (Herbert zur Nedden, 2000)

- Unter RAM 4.x ist das relativ einfach zu lösen. Schalte einfach die Funktionstasten-Tabelle Null (=keine F-Tasten) ein. Dann liefern die Tasten ihren Tastatur-Code direkt und unverfälscht, d.h. F1 liefert \$80 (=80 hex), usw. Was die Cursor-Tasten liefern: siehe RAM 4-Handbuch, Anhang C.
- Ohne RAM 4.x müßt Du das angekommene Zeichen mit dem Inhalt der Tastatur-Tabelle (siehe Info 4) vergleichen. Ist jedoch Memotechs FKEY aktiv, geht dies in die Hose.
- Dritte Möglichkeit: siehe unten meinen Artikel 'Cursortasten-Abfrage'.

A s s e m b l e r: Cursortasten-Abfrage**Cursortasten-Abfrage**

(Herbert zur Nedden, 2000)

Gelegentlich benötigt man eine Tastaturabfrage der Cursor-Tasten mit/ohne Shift sowie der Home-Taste ohne Shift. Da insbesondere beim Original-Memotech diese Tasten dümmlich belegt sind, und daher i.a. mittels Patch unter CP/M erst einmal auf eine vernünftige Belegung umgesetzt werden, und unter RAM 4.x wiederum anders belegt sind, da sie dort zu Funktionstasten geworden sind, muß die Abfrage gegen solche Änderungen immun sein.

Diese hier aufgezeigte Abfrage läuft auf allen Memotechs, bei denen die Tastaturtabelle da ist, wo sie hin gehört, insbesondere tut sie unter RAM 4.x!

Die Tastaturabfrage muß über 0ffd3h erfolgen, damit jegliche Funktionstasten-Decoder und andere Fiesematenten nicht dazwischenfunken.

KbdMove tut folgendes:

Taste	Wirkung	Rückgabe im Akku
Pfeil-Links	dec (EckeX1)	Zero, no Carry
Pfeil-Rechts	inc (EckeX1)	Zero, no Carry
Pfeil-Hoch	dec (EckeY1)	Zero, no Carry
Pfeil-Runter	inc (EckeY1)	Zero, no Carry
Shift-Pfeil-Links	dec (EckeX2)	Zero, no Carry
Shift-Pfeil-Rechts	inc (EckeX2)	Zero, no Carry
Shift-Pfeil-Hoch	dec (EckeY2)	Zero, no Carry
Shift-Pfeil-Runter	inc (EckeY2)	Zero, no Carry
Home	--	NonZero, no Carry
sonst	--	Carry (= Fehler)

Hier nun das Programm selbst:

```

KbdIn          equ      0ffd3h

EckeX1:        dw      0
EckeY1:        dw      0
EckeX2:        dw      0
EckeY2:        dw      0

KbdMove:       ; EckeX1, ..., EckeY2 mit Pfeilen und Shift bewegen

KbdMoveSLeft:  ld      h1,0f226h      ; Shift-Left
               cp      (h1)
               jp      nz,KbdMoveSRight
               ld      h1,(EckeX2)
               dec     h1
               ld      (EckeX2),h1
               xor     a              ; Zero = NoError
               ret

KbdMoveSRight: ld      h1,0f225h      ; Shift-Right
               cp      (h1)
               jp      nz,KbdMoveSUp
               ld      h1,(EckeX2)
               inc     h1
               ld      (EckeX2),h1
               xor     a              ; Zero = NoError
               ret

```

A s s e m b l e r: Cursortasten-Abfrage

```

KbdMoveSUp:   ld      h1,0f227h      ; Shift-Up
               cp      (h1)
               jp      nz,KbdMoveSDown
               ld      h1,(EckeY2)
               dec     h1
               ld      (EckeY2),h1
               xor     a                ; Zero = NoError
               ret

KbdMoveSDown: ld      h1,0f223h      ; Shift-Down
               cp      (h1)
               jp      nz,KbdMoveLeft
               ld      h1,(EckeY2)
               inc     h1
               ld      (EckeY2),h1
               xor     a                ; Zero = NoError
               ret

KbdMoveLeft:  ld      h1,0f1d5h      ; Left
               cp      (h1)
               jp      nz,KbdMoveRight
               ld      h1,(EckeX1)
               dec     h1
               ld      (EckeX1),h1
               xor     a                ; Zero = NoError
               ret

KbdMoveRight: ld      h1,0f1d4h      ; Right
               cp      (h1)
               jp      nz,KbdMoveUp
               ld      h1,(EckeX1)
               inc     h1
               ld      (EckeX1),h1
               xor     a                ; Zero = NoError
               ret

KbdMoveUp:    ld      h1,0f1d6h      ; Up
               cp      (h1)
               jp      nz,KbdMoveDown
               ld      h1,(EckeY1)
               dec     h1
               ld      (EckeY1),h1
               xor     a                ; Zero = NoError
               ret

KbdMoveDown:  ld      h1,0f1d2h      ; Down
               cp      (h1)
               jp      nz,KbdMoveHome
               ld      h1,(EckeY1)
               inc     h1
               ld      (EckeY1),h1
               xor     a                ; Zero = NoError
               ret

KbdMoveHome:  ld      h1,0f1d3h      ; Home
               cp      (h1)
               jp      nz,KbdMoveErr
               ld      a,0ffh
               or      a
               ret                    ; NonZero, NoCarry = No Error, Home

KbdMoveErr:   xor     a
               scf                    ; Zero, Carry = Error
               ret

```

T u r b o - P a s c a l: Cardinals und Zeileneditor**Turbo-Pascal und CARDINALS**

(Herbert zur Nedden, 2000)

Wer kennt es nicht, das Problem unter Turbo-Pascal (TP) mit echten 16-Bit-Zahlen ohne Vorzeichen (also z.B. 16-Bit-Adressen) arbeiten zu wollen. Integer-Zahlen von TP sind leider keine solchen, da das oberste Bit als Vorzeichen-Bit verwendet wird, was bedeutet, daß zwar $\$0010 = 16$, jedoch $\$8000 = -1$ und nicht $32786 (= 8 * 16^3)$ ist. Turbo-Modula ist da etwas praktischer: es kennt den Typ Cardinal, der genau das gewünschte darstellt.

Man kann zwar mit Integer arbeiten, muß aber immer das Vorzeichen mit beachten, um zu erkennen, daß $\$6000 < \8000 ist. I.a. wird dieser Misere durch die Verwendung von Real-Zahlen abgeholfen, die bekanntlich Werte größer als $\text{MaxInt} = \$7fff = 32787$ annehmen können. Da es andererseits gelegentlich erforderlich ist, diese Zahl als echten 16-Bit-Wert zu speichern, bleibt die Frage, wie ich eine 16-Bit-Zahl in eine Real-Zahl hineinbekomme, und wie die Real-Zahl wieder zurück in 16 Bit.

Es ist naheliegend, einen Varianten-Record für diese Tätigkeit zu definieren, mit dem die Arbeit mit Cardinals wie folgt von Statten gehen kann:

```
var Cardinal = Record
    case boolean of
    true: (Wort: Integer);
    false: (Low,High: Byte);
    end;
    Daten: Integer; { das hier ist meine 16-Bit-Zahl um die es geht }
    Zahl: Real; { und mit diesem Feld rechne ich }
begin
    Cardinal.Wort := Daten;
    Zahl := 256.0 * Cardinal.High + Cardinal.Low;
    { nun rechne ich mit Zahl }
    Cardinal.High := trunc(Zahl / 256.0);
    Cardinal.Low := trunc(Zahl - 256.0 * Cardinal.High);
    Daten := Cardinal.Wort;
end;
```

Hinweis: Es sollte sichergestellt werden, daß Zahl eine 16-Bit-Zahl ohne Vorzeichen darstellt, d.h. zwischen 0 und $65535 (= 16^4 - 1)$ liegen muß.

Zeileneditor unter Turbo-Pascal

(Herbert zur Nedden, 2000)

Unter RAM 4.5 gibt es den RAM-Jump EDITOR, der einen Zeileneditor bereitstellt, mit dem nicht nur ein String editierend eingegeben, sondern auch eine bestehender String editiert werden kann. Fragt sich nur 'WIE?'.

Hier die Lösung:

```
var EditorString = Record
    Len: Byte; { Max-Laenge (muss <= 128 sein) }
    Str: String[128]; { der String mit akt. Laenge }
    end;
begin
    EditorString.Len := 128; { oder was auch immer als Max-Laenge gewünscht }
    EditorString.Str := 'Dies ist der alte Inhalt';
    inline ($11/EditorString/ { LD DE,EditorString }
            $cd/>$f95a); { CALL EDITOR }
    { fertig }
end;
```

C P / M: ZCPR2

GET von ZCPR2

(Herbert Oppmann, 8520)

Wie ja schon bemerkt (Info 28 Seite 6) liefert bei unserem heißgeliebten ZCPR2 DIR nach GET einen beträchtlichen Müll, der erst wieder aufhört zu erscheinen, wenn man ^C drückt (warmboot). Nun hab ich mich mal über die Source gemacht und festgestellt, daß der Fehler darin liegt, daß die DMA-Adresse von GET verstellt wird, aber nicht mehr auf den Defaultwert von 0080H (TBUFF) zurück! Dir geht nun aber davon aus, daß die DMA-Adresse default ist. DIR läßt sich vom BDOS einen Eintrag nach dem anderen geben, der vom BDOS fälschlicherweise in die TPA kopiert wird. DIR sieht jetzt in TBUFF nach - tja und was steht da wohl? Meistens ein Directory-Sektor von anno dunne mal. Dort sind maximal 4 Dateien eingetragen, so daß DIR nun immer nur diese vier anzeigt, und zwar sooft, wie Dateien anzuzeigen sind (hängt von der angegebenen Maske und dem Inhalt des betrachteten Directories ab). Nun hab ich mir mal näher angeguckt, wo überall die DMA-Adresse verändert wird und wo überall davon ausgegangen wird, daß sie auf 0080H steht. Und dabei hab ich festgestellt, daß der Autor von ZCPR2 sich anscheinend nicht entscheiden konnte, ob er

- a) immer nach dem Verändern der DMA-Adresse dieselbe wieder auf 0080H stellt oder
- b) immer vor den Stellen, an denen die DMA-Adresse 0080H sein sollte.

Manchmal macht er's so, dann so - dann wieder doppelt gemoppelt, tja - und manchmal vergisst er's! Würg!

Anm. d. HzN.: Hattu RAM 4.5, dann muttu auch nächsten Artikel lesen und nicht gleich Herbert's Patch einbauen!

Um das ganze zu Bereinigen, müßte man sich zunächst mal klar werden, ob man nach Schema a) oder b) verfahren möchte, und dann an allen möglichen Stellen im Source ändern. Da das ein Neuassemblieren erfordert, welches ich nicht allen Leuten zumuten wollte, mußte ein möglichst einfacher (aber zuverlässig funktionierender!) Patch her. Ich habe festgestellt, daß genau in drei Fällen vergessen wird, die DMA-Adresse wieder auf 0080H zu setzen, und zwar

- a) bei GET allgemein,
- b) bei SAVE, wenn ein Fehler auftritt,
- c) bei GET und COM, wenn ein Fehler auftritt.

Mit COM meine ich das Laden einer COM-Datei in die TPA. Nun sind die Fälle b) und c) recht unwahrscheinlich, weshalb sich nur Fall a) verriet. Schauen wir uns das mal ein wenig genauer an.

zu Fall b)

```
SAVE:      ...
           (Schreiben. Wenn Fehler: Sprung nach SAVE3)
           ...
           JR   NZ,SAVE4
SAVE3:     CALL PRNLE      ;PRINT 'NO SPACE' ERROR
SAVE4:     JP    DEFDMA
```

(DEFDMA setzt die DMA-Adresse auf den Wert 0080H)

Das sieht doch ganz gut aus, oder? Auf den ersten Blick ja, aber der Witz ist, daß CALL PRNLE nicht mehr zurückkommt!

C P / M: ZCPR2

zu Fall a)

GET und COM rufen MLOAD auf. Bei COM wird vor dem Sprung nach 0100H, also dem Starten des geladenen Programms, noch einmal DEFDMA aufgerufen, somit ist alles Paletti. GET tut das dummerweise nicht.

zu Fall c)

```
MLOAD:    ...
           (Lesen. Wenn Fehler: Sprung nach PRNLE)
           ...
           JP  Z,DLOGIN
;
;LOAD ERROR
;
PRNLE:    CALL PRINTC
           DEFB 'Fu1','1'+80H
           CALL DLOGIN          ;RESTORE CURRENT USER/DISK
           JP  RESTRT          ;RESTART ZCPR
```

Gleicher Fall wie Fall b)!

Praktischerweise wird PRNLE jedoch nur von SAVE und MLOAD verwendet, so daß ein Einbau von CALL DEFDMA schon mal Fall b) und c) erschlägt. Wenn jedoch kein Fehler in MLOAD auftrat, dann wird JP Z,DLOGIN ausgeführt und wenn MLOAD von GET aufgerufen wurde, dann --> Sierra Nevada. Also muss auch wenn MLOAD klappte, noch vor dem Verlassen von MLOAD einmal DEFDMA aufgerufen werden. Bei COM ist das natürlich jetzt doppelt gemoppelt, aber was soll's. Als besondere Schwierigkeit des Patches erweist sich, daß nach PRNLE, das ist glücklicherweise das Ende von ZCPR2, nur noch 5 Byte frei sind.

Was hab ich also getan: aus JP Z,DLOGIN ein JR Z, .. gemacht (bringt ein Byte Platz), dann PRNLE um ein Byte nach vorne verschoben und die Referenzen auf PRNLE in SAVE und MLOAD (je eine) korrigiert. Dann den erwähnten JR Z,.. sowie den CALL DLOGIN hinter PRNLE verbogen. Dort folgende Routine untergebracht:

```
BUG:      CALL DLOGIN
           JP  DEFDMA
```

Das braucht genau die 6 (5+1) freien Byte auf und löst hoffentlich das Problem. Eine Änderung im Speicher ist nicht sinnvoll, da ja bei jedem ^C das ZCPR2/P2DOS von den Systemspuren der Ramdisk geholt wird. Auch dort ist ein Patch wenig sinnvoll, denn der müßte ja jedesmal nach dem Booten durchgeführt werden. Also gehen wir gleich an's Eingemachte: die Systemspuren auf der Bootdiskette.

ACHTUNG:

Die folgende Anleitung ist nur für unverändertes ZCPR2 (MOD 0.3 OF THE RELEASE VERSION) und eine Systemgröße von 58k (ist ja fast ein Standard) !!! Bei einer anderen Systemgröße sind einige Bytes anders !!! Das ist mir hier aber zu kompliziert darzustellen, deshalb anrufen, falls Probleme.

Anm.d.HzN.: Hattu RAM 4.5, dann hattu auch nicht ZCPR2 Mod 0.3!

Zunächst nehmen wir ein leere, formatierte Diskette und duplizieren unsere Bootdiskette inklusive Systemspuren (entweder mit COPYD, oder mit WUSCH und XSYS oder ähnlichem). Dann stellen wir die Original-Bootdiskette sicher beiseite und die Kopie stecken wir in ein Laufwerk (im Folgenden wird B: benutzt). Jetzt werfen wir eine Disk-Utility an, z.B. das mitgelieferte DU2:

C P / M: ZCPR2

Eingaben	Kommentar
du2/	starten. / bedeutet Return-Taste
lb/	Laufwerk b:
t0/	Track 0 (logische Spur 0)
s14/	Sektor 14 (dezimal). Sektor wird gelesen
d/	Sektor anzeigen. Byte 01 sollte EE sein.
ch01 ed/	change hex (hexadezimal ändern) Hiermit wird CALL PRNLE in SAVE korrigiert.
w/	Zurückschreiben
s17/	Sektor 17. Letzter ZCPR2-Sektor
d/	Anzeigen. Byte 55 sollte 18 sein
ch55 17/	JR NC,PRNLE wird korrigiert
ch6b 28 0d cd 9a cc	46 75 6c ec cd fa d2 c3 9f cb cd 6d ce c3 b5 cc/ Ab Byte 6b sollte vorher gestanden haben: CA 6D CE CD 9A CC 46 75 6C EC CD 6D CE C3 9F CB
w/	Zurückschreiben
x/	Verlassen
n	braucht kein warmboot

Jetzt kann von dieser Diskette mal ein Bootversuch gewagt werden. Wenn alles klappt, dann mal GET und hinterher DIR probieren. Sollte jetzt keinen Müll mehr liefern.

OK, ein Durchforsten des Codes und neu Assemblieren wäre besser gewesen, aber das überlasse ich den für neue Betriebssysteme zuständigen Leuten, denn ich will nicht auch noch zur Vielfalt der Systeme beitragen. Falls ZCPR3 beim nächsten größeren RAM-Update nicht Verwendung findet, empfehle ich dann doch eine Überarbeitung von ZCPR2.

Zum ZCPR2 unter RAM 4.5, d.h. dem geänderten: (Herbert zur Nedden, 2000)

Ich bat Herbert, seinen o.g. Artikel auch gleich noch einmal an dem für RAM 4.5 veränderten ZCPR2 revue passieren zu lassen, damit ich das ganze gleich vom Fachmann aufbereitet erhalte und auch verstehe. Die Antwort von Herbert hat mich etwas verblüfft, meine Brechstangen-Methode aufgezeigt (peinlich) und gleich noch die eine oder andere gute kritische Anmerkung gebracht. Danke Herbert!

Zum ZCPR2 unter RAM 4.5, d.h. dem geänderten: (Herbert Oppmann, 8520)

Nachtrag:

HZN schickte mir RAM 4.5, dort ist ein neu assembliertes ZCPR2 dabei. Dieses ist für RAM 4.5 unbedingt zu verwenden, und obiger Patch ist mit diesem ZCPR2 nicht mehr nötig. Ein "CALL DEFDMA" wurde hier so eingebaut, daß es vor jeder Kommandozeilen-Eingabe aufgerufen wird. Mit dieser Brechstangen-Methode ist man auf alle Fälle auf der sicheren Seite. Trotzdem wollte ich euch obige Ausführungen nicht vorenthalten (wenn ich mir schon mal die Mühe mache, etwas fürs Info zu schreiben ...).

C P / M: ZCPR2

Die Korrektur mit "CALL DEFDMA" wurde nicht beim DIR-Befehl, sondern am Beginn jeder Eingabe eingebaut. Das ist zwar eine Brechstangenmethode, aber funktioniert wohl. Etwas anderes ist dagegen, daß DiJey anscheinend die DMA-Adresse verändert, ohne sie wieder zu restaurieren. Jedenfalls ergeben sich mit DiJey die vom GET-Befehl her bekannten Symptome, und das trotz meiner Korrektur (von deren Korrektheit ich nach wie vor überzeugt bin). Bei Verwendung des von euch modifizierten ZCPR2 bemerkt man wegen der Brechstangenmethode das zwar nicht mehr, aber erstens finde ich es nicht gut, einen Mangel, der bei DiJey beheimatet ist, in ZCPR2 auszumerzen, und zweitens stellt sich die Frage, wie andere Programme (z.B. NewWord) wohl darauf reagieren, wenn asynchron (soll heißen, im laufenden Betrieb) die DMA-Adresse verstellt wird? Mir gefällt das gar nicht, vor allem, weil sich in der Doku kein Hinweis auf diese Gefahrenquelle befindet, und die Folgen doch beträchtlich sein können.

Anm.d.HzN.: Olaf warnt durchaus in seiner Doku zu DiJey, daß man bei Schreibzugriffen auf ein Laufwerk mit offenen Dateien (insbesondere also auf das Laufwerk, auf dem der aktuelle NewWord-Text ist) mit guter Chance seine Dateien zerschießt!

Wir haben vor, die BDOS-Variablen, soweit es Sinn macht zu retten, wozu sie allerdings erst einmal im COMMON stehen müssen, da nicht immer das BDOS im Speicher steht, wenn ein KLICKer aktiviert wird. Ich hoffe, Du hältst es bis RAM 5.x aus - unter dem jetzigen RAM 4.5 wäre eine sauberere Arbeit von DiJey nur mit sehr fiesen und auch nicht sicheren Methoden machbar. Denn wie soll DiJey BDOS-Variablen retten, wenn er nicht einigermaßen vernünftig feststellen kann, ob das BDOS überhaupt aktiv ist? Die offizielle Kennung des BDOS gleich zu seinem Beginn wird ja von 50% der Stacks überschrieben!

Die Benutzung des BDOS durch KLICKer, ohne daß die BDOS-Variablen gerettet werden, führt dazu, daß KLICKer nicht zwischen "search first" und "search next" aktiviert werden dürfen, sonst kommt Mist heraus. Aber - wie letzten Herbst mit Dir (=HzN) in Hannover besprochen: die Wahrscheinlichkeit ist gering, da wohl kaum jemand eine Diskettenoperation unterbricht um einen KLICKer aufzurufen, der das BDOS benutzen soll und wenn doch, dann ist er selbst dran schuld, und die Folgen halten sich sowieso in Grenzen (es wird halt was anderes oder nix gefunden). Hier jedoch ist das anders: die Wahrscheinlichkeit, ein Programm zu unterbrechen, das die DMA-Adresse weiterverwenden wollte, scheint mir recht groß zu sein, und die möglichen Folgen sind auch nicht zu unterschätzen. DiJey scheint auch sonst noch eine Macke zu haben: wenn eine neue Disk eingelegt wird, kann man auch drauf einloggen, aber das ändert nichts daran, daß diese Disk dann schreibgeschützt ist und immer noch der freie Platz der vorherigen Diskette angezeigt wird.

Anm.d.HzN.: Das ist gewollt! Ein Einloggen macht KEIN Disk-Reset, und daher ist die Scheibe schreibgeschützt, falls sie gewechselt wurde. Das ist übrigens unter CP/M auch so: ohne ^C oder entsprechende Maßnahmen ist eine gewechselte Scheibe schreibgeschützt, egal ob sie eingeloggt wird, oder nicht! Der freie Platz wird aus dem Allocation-Vector, d.h. der Blockbelegungstabelle ermittelt, und die wird nur nach einem Disk-Reset eingelesen! Daher muß Du unter DiJey ein Disk-Reset mittels des Z-Kommandos für das gewünschte Laufwerk explizit durchführen!!! Eine Automatik an dieser Stelle wäre recht gefährlich! Würde nämlich DiJey immer beim Einloggen ein Disk-Reset durchführen, gäb's bei offenen Dateien leicht recht heiße Resultate.

C P / M: MENU

MENU

(Herbert Oppmann, 8520)

In Info 28 Seite 5 schreibt Peter Würfel, daß MENU.CPR maximal 16k lang sein darf, da sonst Absturz. In Info 30 Seite 20/21 schreibt Hans Gras dazu, daß es an der SYSLIB lag:

"Beim Lesen eines Record wird keine Fehlermeldung gegeben, wenn wir am Ende des Files sind. Der letzte Record wird dann immer wieder gelesen - bis zum Überschreiben des ZCPR2 und P2DOS. Dies gab einen schönen Absturz."

Tja, das Erscheinungsbild kann ich bestätigen, die Ursache liegt meiner Meinung aber wo ganz anders (und deshalb kann auch Peter Würfel ganz beruhigt sein, daß Hans' korrigierte Version bei ihm nicht läuft). Sorry Hans! Aber in meiner Source (CLUB 019), MENU Version 1.4, sieht es so aus, daß MENU zuerst die ganze Datei auf einmal hinter sich in die TPA lädt und dabei auf das Überschreiben des ZCPR2 achtet (!). Selbst wenn also der letzte Record oft gelesen wird, dürfte nix passieren. Es ergibt sich aus diesem Vorgehen allerdings auch, daß MENU.CPR niemals größer als (64k - Größe MENU - Größe ZCPR2/P2DOS/Residenter Teil des BIOS) sein darf, da die Datei sonst kommentarlos abgehackt wird. Das dürften aber so ungefähr 40k sein, es sollte also bei 16k noch nicht Schluß sein - wenn sich da nicht ein übler Fehler eingeschlichen hätte! Wenn man sich die Source ansieht, stellt man fest, daß die zweite Hälfte des FCB (File Control Block) für MENU.CPR doppelt verwendet wird (Label "scratch"). Nun die Idee war wohl, Speicher zu sparen (ist ja auch ok, denn nach dem Einlesen von MENU.CPR wird der FCB nicht mehr gebraucht und der Platz steht dann für diverse Variablen zur Verfügung). Eine Aufstellung dieser Variablen befindet sich ganz am Ende der Source ("wait equ scratch" usw.). Und unter diesen Variablen befindet sich ein schwarzes Schaf: "mladr". Und das geht so:

- o FCB initialisieren (alles hinter Name mit 0 auffüllen)
- o Datei suchen und öffnen
- o Ladeadresse ermitteln (ergibt sich aus Ende von MENU)
 - und nach "mladr" speichern - Moment! der FCB wird doch noch gebraucht!!!

Da bei 2k Blockgröße in einen physikalischen Extent zwei logische Extents a 16k passen, kracht es hier noch nicht, denn "mladr" befindet sich in der zweiten Hälfte der Blocknummern. Doch der Krimi geht weiter:

- o Records von Datei lesen, bis Datei zu Ende oder Lesefehler oder kein Platz mehr unterhalb von ZCPR2.

Jetzt schlägt P2DOS zu: wenn ein neuer logischer Extent geöffnet wird, also beim Überschreiten von 16k, wird der physikalische Extent neu in den FCB kopiert --> Adieu "mladr"! Dort steht jetzt 0000H oder, wenn MENU.CPR länger ist, eine Blocknummer.

- o Datei schließen

Das klappt auch noch ganz gut.

- o Gelesene Datei scannen

Und jetzt kracht's! "mladr" enthält Käse, und MENU versucht nicht nur ab "mladr" zu lesen, sondern löscht jeweils das 7te Bit!!! Und wenn dieses "mladr" in das P2DOS oder BIOS oder gar MENU selbst zeigte, dann gute Nacht!

C P / M: M E N U / C A T

Abhilfe: "mladr" wo anders hin legen (diese 2 Byte mehr kosten nicht viel). Falls man nicht neu übersetzen will: Patchen. "mladr" wird nur dreimal verwendet, deswegen ist es ganz einfach. In SINSFORM ist am Ende noch Platz reserviert für Erweiterungen. Davon knapse ich mal schnell zwei Byte ab für ein neues "mladr". Das Highbyte der Adresse bleibt sogar gleich, so daß nur drei Byte zu ändern sind.

Für MENU Version 1.4:

Eingaben	Kommentar
zsid menu.com/	Debugger starten. / bedeutet Return-Taste DDT oder DDTZ oder MONI2 oder ... gehen auch
s1dc/6c./	dort steht jeweils 8CH, da "mladr" = 018CH
s21d/6c./	neue Adresse 016CH
s269/6c./	
^C	Debugger verlassen
save 8 NMENU.COM/	neues MENU abspeichern

Für MENU Version 1.5: (hier ist alles ein Byte nach hinten verschoben, ansonsten aber gleich, was "mladr" betrifft. Bei MENU 1.5 steht ganz am Anfang bei Adresse 0100H eine 0 = ein NOP. Was soll das ??).

Eingaben	Kommentar
zsid menu.com/	
s1dd/6d./	
s21e/6d./	
s26a/6d./	
^C	
save 43 NMENU.COM/	Warum so riesig? Schleppt wohl die ganze SYSLIB mit sich rum?

Nachtrag:

HZN schickte mir MENU Version 2.0 zum Testen. Diese Version ist ziemlich stark verändert (verbessert?). Jedenfalls tritt oben beschriebener Fehler nicht auf und somit ist für diese Version auch kein Patch erforderlich.

CATalog-Programme: Tips und Erfahrungen

(Peter Würfel,7262)

Das Problem kennen alle: Ich weiß den Namen (auf jeden Fall in etwa) eines Files (=Datei, Anm.d.HZN), weiß aber nicht mehr, auf welcher Diskette es sich versteckt. Die Lösung ist eine Gruppe von Programmen, die man auf den CLUB-Disketten 032 und 041 finden kann: Ein Programm (NEWCATx.COM) dient dazu, die Inhaltsverzeichnisse von Disketten in eine Datei einzulesen, andere Programme (CATx.COM, OLDCAT.COM, PUTCAT.COM) übernehmen die Aufgabe, diese Datenbank sortiert, nach gewünschten Kriterien wieder zugänglich zu machen. Die ursprünglichen Katalog-Programme findet man auf CLUB.032 zusammen mit erweiterten Versionen von Hans Gras. Auf CLUB.041 befindet sich eine neue Version von Hans Gras.

Die drei NEWCAT-Programme unterscheiden sich wie folgt:

- NEWCAT.COM (CLUB.032) liest nur das Directory von User 0, andere User können nicht eingelesen werden.
- NEWCAT4.COM (CLUB.032) kann auch das Directory anderer User-Bereiche einlesen, doch immer nur **ein** User-Bereich; dazu muß **in jedem** User ein Dateinamen-Eintrag der Form '-<name>.<Nr>' vorhanden sein.
- NEWCAT.COM (CLUB.041) liest auf einen Schlag **alle** User ein.

C P / M: CAT

Alle drei Programme schreiben ihre Daten in MAST.CAT. Das Programm c) speichert jedoch auch den User-Bereich, in dem ein File abgelegt ist. Das von den Programmen a) und b) erzeugte MAST.CAT unterscheidet sich also im Aufbau von dem durch c) erzeugten MAST.CAT.

Mit den Programmen CATx.COM kann man nun die in MAST.CAT abgelegten File-Namen insgesamt oder teilweise auf den Bildschirm schicken. Die CATx.COM-Programme von CLUB.032 zeigen den User-Bereich nicht an, CAT.COM von CLUB.041 zeigt auch an, in welchem User sich ein File befindet. Und da kann es zu Problemen kommen. CAT.COM von CLUB.041 erfordert nämlich ein MAST.CAT, in dem die User abgelegt sind, ein MAST.CAT also, das mit NEWCAT.COM von CLUB.041 erzeugt wurde. Wenn man CAT.COM (CLUB.041) auf ein altes 'MAST.CAT' losläßt, gibt es Bildschirmmüll, wenn man versucht dem NEWCAT (CLUB.041) ein 'altes' MAST.CAT zum Einschreiben der Daten vorzusetzen, werden abenteuerliche Daten eingelesen. Was also tun, wenn man vom 'alten' CAtalog-System ohne User-Einträge auf das 'neue' System mit User-Einträgen umsteigen möchte? Ist es notwendig alle katalogisierte Disketten noch einmal neu zu bearbeiten? "Keine Panik" (D.Adams, 'Per Anhalter durch die Galaxis'), man muß nicht, sondern:

- 1) NW aufrufen und MAST.CAT im **NonDoc**-Modus laden.
- 2) Ans Ende von MAST.CAT gehen, und mit Suche-Ersetze an alle Zeilen ein ';0' anhängen:

<CTRL><Q><A>	<i>Suche-Ersetze aufrufen</i>
Find what? ^M<RET>	<i>Eingabe: <CTRL><P>+<CTRL><M></i>
Replace with? ;0^M<RET>	<i>^M wie oben</i>
Option(s)? nrb<RET>	<i>nix fragen ob, Rest der Datei, Rückwärts</i>
- 3) Nun müssen noch die 'FREE-Zeilen' geändert werden:
 Eine FREE-Zeile des 'alten' MAST.CAT sieht z.B. wie folgt aus:
 ++++0k.FRE,KX.411 oder
 +++110k.FRE,TEST.001
 Im 'neuen' MAST.CAT sehen diese Zeilen wie folgt aus:
 ++++0kB.FRE,KX.411;0
 +++110kB.FRE,TEST.411;0
 Die Sache mit dem ';0' haben wir ja schon unter 2) erledigt, bleibt also noch, alle 'k' durch 'kB' zu ersetzen (da es sonst in MAST.CAT keine kleinen 'k's gibt, kann das mit Suche-Ersetze automatisch geschehen); was bleibt, ist in alle FREE-Zeilen ein '+' hinzuzufügen, das geht halt nicht mit Suche-Ersetze, sondern Zeile für Zeile.
- 4) 'Neues' MAST.CAT speichern und damit arbeiten!

Wen es stört, daß nun alle Files auf User 0 angegeben sind, der soll sich halt die Arbeit machen, alle Disketten mit mehreren Usern neu zu katalogisieren; ich hab mir das gespart und schiebe nur mal von Zeit zu Zeit eine Diskette für'n Update ins CAtalog-Programm.

Was ich noch zu sagen hätte..(zu den CATsen auf CLUB 041)

1. Vielen Dank an Hans für die verbesserte CAT-Version!
2. Vielen Dank an Herbert für sein DOC zum CAtalogssystem auf CLUB.041!
3. **NEWCAT** funktioniert auch, wenn man **nur ein** Laufwerk hat. Wenn NEWCAT aufgerufen wird, fordert es zuerst die zu katalogisierende Diskette, liest das neue Directory ein und fragt dann nach der Diskette mit dem MAST.CAT-file.
3. Speicherstelle 106 Hex zu patchen, um **NEWCAT.COM** bzw. **CAT.COM** zu sagen, wo es sein MAST.CAT zu suchen hat, reicht nicht. NEWCAT.COM bzw. CAT.COM finden MAST.CAT nur, wenn das gewünschte Laufwerk im Pfad liegt. (Also wenn nötig mit PATH das Laufwerk, auf dem MAST.CAT ist, 'einreihen'.)

C P / M: CAT

4. **NEWCAT** braucht viel Speicherplatz: Wenn ein neues Directory eingeladen wird, sind anschließend zwei MAST-files auf der Diskette: MAST.CAT und MAST.BAK. Man könnte also meinen, es genüge ein max. Speicherplatz, der etwas größer als doppelt so groß wie MAST.CAT ist. Dem ist jedoch nicht so: Man braucht den **dreifachen** Speicherplatz! NEWCAT läßt nämlich zuerst die Files MAST.CAT und MAST.BAK unangetastet und legt, um das neue Directory einzusortieren, eine Datei MAST.NEU an. Erst wenn dieser Vorgang fehlerfrei abgelaufen ist, wird MAST.CAT zu MAST.BAK (das alte MAST.BAK geht flöten) und MAST.NEU wird zu MAST.CAT. Da ich mein MAST.CAT auf meinem Schneider-LW mit mageren 170kB habe, muß ich also (mein MAST.CAT hat 70kB) zuerst MAST.BAK löschen (era..) und kann dann erst NEWCAT aufrufen. Da wäre es doch besser, das Programm würde nur das MAST.CAT unangetastet lassen und mit seinem MAST.NEU das etwaige MAST.BAK überschreiben. Datensicherheit wäre dadurch doch auch ausreichend gewährleistet.
5. Und noch ein **Verbesserungsvorschlag**: CAT.COM sollte immer dann auf einen Tastendruck warten, wenn eine Bildschirmseite voll ist. (Und natürlich verraten, in welcher Speicherstelle der Zeilenzähler zu finden ist, damit ich mir meine 28 Zeilen reinpatchen kann (es soll ja immer noch Leute geben, die nur 24 (oder wenns hochkommt (wem oder was) 25)) Zeilen haben) (eine Klammerorgie für BOOLSche Freaks). Dann kann man auch auf PUTCAT (s.u) verzichten, gibt's doch im KLICK das wunderschöne Hardcopy.
6. Und wenn ich schon **Wünsche** äußere: NEWCAT sollte immer dann, wenn es die Erweiterung .LBR entdeckt, auch die in dieser Library 'versteckten' Dateinamen in den Katalog übernehmen. In CAT könnte dann ja bei diesen Namen in der User-Spalte neben der Ziffer für den User ein 'L' erscheinen.
Anm.d.HzN.: Olaf Krumnow hat die Library-Zugriffsroutinen in Assembler, d.h. wer auch immer sich diesem Wunsche erbarmt, dem können wir die Suche nach diesen Routinen von diversen PD's ersparen.
7. Und noch'n **Wunsch**: NEWCAT sollte auf dem Bildschirm den Diskettenamen der Diskette angeben, die gerade bearbeitet wird (z.B.: 'In Arbeit: -util.245')
8. **PUTCAT**: Wer seinen Katalog ausgedruckt haben will (will da wer?), der muß PUTCAT.COM und MASTCAT.COM und ausreichend Speicherplatz für das NWMAST.CAT **auf der gleichen** Scheibe (Disk oder RAM-Disk) haben. PUTCAT könnte doch wenigstens fragen (meinetwegen auf Holländisch), wo er das MAST.CAT suchen soll, wenn er's (warum eigentlich 'er' und nicht 'sie' oder 'es'? Anm. des Über-Ich) nicht findet. Und warum braucht man dazu 30k? (Oh Du ahnungsloser Engel Du, Anm. des Über-Ich) Kleiner wär feiner? (...)
9. **OLDCAT**: auch etwas groß (28k), dafür kann aber MAST.CAT auch auf einem anderen LW als OLDCAT liegen.
10. **RAUSCAT**: ein Programm dieses Namens (Es gibt ihn noch! Den Genitiv!) gibt es nicht, aber es soll vorkommen, daß eine Diskette nicht mehr vorhanden ist (logisch oder physikalisch), im CATALog-System aber immer noch gespeichert ist. Einen Befehl: 'Lösche alle Einträge der Diskette <-test.999>' gibt es nicht. Und braucht man auch nicht, wenn man folgendes tut:
 - leere Diskette mit dem Disketten-Namen der zu löschenden Diskette versehen.
 - diese Diskette NEWCAT vorsetzen: NEWCAT wird alle bisher unter diesem Diskettenamen gespeicherten Files löschen und nur ein FREe-Eintrag wird in MAST.CAT zurückbleiben.
 - die FREe-Zeile in MAST.CAT entfernen (NewWord, NonDoc)**Anm.d.HzN.:** Aber schön wäre er wohl doch, oder ?

Jetzt fehlt nur noch das Programm, das mir sagt, wie das Programm heißt, das ich such, aber dessen Name mir nicht einfällt..(hä?)!

C P / M: NewWord & ZEX**NewWord und ZEX - Neues und Korrektur Info 26-5f**

(Peter Würfel, 7262)

Da ich immer wieder SuperCalc-Listen (in festem Format) mit NW in ein und der selben Art und Weise umarbeiten mußte, machte ich mir die Arbeit, ein entsprechendes Zex-File zu schreiben, das mir diese regelmäßig wiederkehrende Arbeit abnehmen soll. (Ganz nebenbei: Bis heute ist unklar, wer da wem Arbeit gemacht bzw. abgenommen hat) Dabei machte ich folgende Erfahrungen:

Anm.d.HzN.: Zu Deiner Frage, wer da wem Arbeit gemacht hat: Naja, Dein NewWord und Du mit Deinen Problemen mir, der ich versuchte Dein langes ZEX-File zu verstehen, damit ich dahinter kam, warum es bei mir (trotz dem es bei Dir nicht lief) lief. Und damit hast Du Dir - und damit Deinem Rechner Arbeit gemacht - weil Du mir das alles sehr schön verständlich zusammengestellt hast, so daß ich das Ganze fast sofort laufen lassen konnte - ich verstand halt nicht, was Dein Problem war - bei mir lief ja alles - bis ich das richtig große .ZEX versuchte. Jedenfalls hat uns die Post den Transport der Disketten abgenommen. Ich muß sagen, daß ich bei dieser Gelegenheit gelernt habe, daß es i.a. schusselig ist mit SUB oder XSUB zu arbeiten, da ZEX viel mehr kann.

Wenn durch ein umfangreiches *.zex **nicht genügend Platz** für das NewWord bleibt, kann dies dazu führen, daß

- entweder NW nach dem Aufruf garnicht gestartet wird (s' Pferdle ((c) SDR) scheut vor dem Hindernis, weil's eh nicht drüberkommt)
- oder NW zwar noch gestartet wird, es aber mehr oder weniger gravierende Probleme gibt:
 - entweder beim Vor- und Zurückblättern; dies führt dann ins Nirwana (selbst <SHIFT><ESC> ist dann tot, es helfen nur noch die zwei berühmten schwarzen, unbeschrifteten Tasten). (Hinterlistiger Gaul: Springt und landet im Hindernis)
 - oder das Programm läuft ohne Absturz, wenn man aber ausdrucken will, kommt die Fehlermeldung '...there is Not enough memory...' (s'Pferdle hät sichs andaschda iberlägd), in diesem Fall kann man ohne Datenverlust aus NW aussteigen.

Daß diese Fehler bei mir auftraten, lag daran, daß ich mein NW sehr speicherplatzhungrig installiert hatte bzw. habe: Mein NW arbeitet nach dem Aufruf sofort mit dem 96*28-Schirm und außerdem hatte ich einen 'unerase-Puffer' von 3000 vorgesehen (habe ihn nun auf 500 verkleinert). Wenn ich den **Platzbedarf von NW** durch entsprechende Installation verkleinere, gibt es keine Probleme mehr!

ZEX verkraftet übrigens ohne Probleme, wenn ich gleich nachdem NW aufgerufen wurde ins **Klick** gehe und das **Bildschirmformat verändere**.

Meine Zex-Files schreibe ich mit NewWord im Non-Document-Modus. Die zwei Möglichkeiten, **Control-Codes** in ZEX unterzubringen, wurden von Uwe Grass und Jan Brederke in Info 26-5f beschrieben (Danke, ihr zwei für euren damaligen Artikel, kam er doch auf eine Anfrage von mir in Info 25 zustande und hat mir damals ein gutes Stück weitergeholfen!)

Control-Codes füge ich durch <CTRL><P>+<CTRL><Buchstabe> (Methode U.Grass) ein, das hat m.e. den (kleinen) Vorteil, daß nicht zwei sondern nur ein Byte gebraucht wird und das ZEX-File kürzer bleibt. Doch da gibt es zwei Ausnahmen:

- **Control-J** (^J,ASCII: 0Ah, Tastenfolge: <CTRL><P>+<CTRL><J>) verkraftet NW nicht. In diesem Fall schreibe ich ins ZEX-File dann 'Dächle'-Zeichen+J (<CTRL>+<J>): ^J (zwei Zeichen).
- **Control-U** (nur, dieser zweite Fall ist für die Praxis ohne Bedeutung)

C P / M: NewWord & ZEX

Ein weiteres Problem betrifft 'Suchen und Ersetzen' (mal wieder) von ZEX aus. Ich beschreibe, was das ZEX-File tun soll:

- nw aufrufen
- 'Test.txt' zur Bearbeitung aufrufen
- alle Doppelpunkte ':' durch '^ER^R' ersetzen (d.h. ich ersetze alle Doppelpunkte durch 'Custom E'+ 'R'+ 'Custom R')
- 'Test.txt' abspeichern und nw verlassen

Doch das klappt nicht vollautomatisch. Was funktioniert, ist folgendes (hier die beiden möglichen ZEX-Files; die Codes, die über <CTRL><P> eingegeben wurden, sind hier unterstrichen, Zeilennummern wurden von mir der Übersichtlichkeit halber hinzugefügt):

	(J.B.-Eingabe)	(U.G.-Eingabe)
1:	^#nw Test.Txt	^#nw Test.Txt
2:	^Qa:	^Qa:
3:	^PR^P	<u>^PR^P</u>
4:	^Ä^Kx	<u>^Ä^Kx</u>

Die rechte Möglichkeit ist fünf Byte kürzer. Das entscheidende ist die Zeile 3: weder ^ER^R noch ^ER^R (das letztere würde man eingeben, wenn man diesen Vorgang 'von Hand' in NW durchführen würde) führen zum Erfolg. Doch wie funktioniert nun das obige Programm. ZEX macht, was es soll, bis es in Zeile 3 ankommt. Dann wartet ZEX auf zwei Eingaben (in meinem Fall <CTRL><E> und anschließend <CTRL><R>) und fährt daraufhin mit der korrekten Abarbeitung des Programms fort.

Das eben gesagte gilt genauso für den umgekehrten Fall: Ich möchte Control-Zeichen im Text durch 'einfache' Zeichen ersetzen; auch in diesem Fall funktioniert nur die Lösung über ^P!

Nun **zwei kleine Tips**, die ich in Zusammenhang mit obigem Problem gefunden habe.

- Wenn ZEX schon nicht vollautomatisch abläuft, so möchte ich doch nicht die ganze Zeit auf den Bildschirm starren und warten, bis meine Eingabe erforderlich ist. Also steht bei mir in Zeile 3 zuerst: ^* (2 Zeichen!!); wenn ZEX hier angekommen ist, meldet es sich durch einen Piepser. Dieser ZEX-Befehl '^*' erzeugt einen Piepton ohne ein Programm zu beeinflussen.
- Und da ich weiß, daß ich (spätestens in drei Tagen) nichts (mehr) weiß, nämlich von dem, was ich eingeben muß, wenn mein ZEX-File in Zeile 3 auf meine Eingabe wartet, finde ich die beiden folgenden (zusammengehörenden) ZEX-Befehle sehr hilfreich: '^<' und '^>' (Wichtig: jeweils 2 Zeichen). Alles, was zwischen den beiden Befehlen steht, auch Bildschirmsteuerzeichen, wird quasi am laufenden Programm vorbei auf den Bildschirm geschrieben.

Meine Zeile 3 sieht somit wie folgt aus:

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

Zum **Zusammenspiel von MENU, ZEX und NW** noch folgender Hinweis: Wenn ich von Menu aus ein *.ZEX-file aufrufe, das wiederum NW zur Bearbeitung eines (Non-)Dokuments aufruft, sollte das *.ZEX-file nicht in NW enden, sondern NW sollte unter ZEX (z.B. mit ^Kx) verlassen werden. Wenn ZEX mitten im Dokument endet, wird die restliche Kommandozeile ins Dokument geschrieben. Somit wird, nachdem NW beendet wurde, MENU nicht aufgerufen, und man findet sich auf der ZCPR2-Ebene wieder.

C P / M: NewWord & ZEX / SUB - ZEX - MENU

Schließlich noch einige kleine Korrekturen zu Info 26-5, TEST.SUB :

- XSUB in Zeile 1 wird nicht gebraucht. (Wird XSUB überhaupt irgendwann zu irgendetwas gebraucht, wenn man ZEX hat?)
Anm.d.HzN.: Da mein SLR-Assembler bei Fehlern bereit ist, eine laufende .SUB-Datei abzubrechen, auf .ZEX jedoch ignorant reagiert, nutze ich dafür grundsätzlich SUB oder XSUB.
- In der vierten Zeile sind zwei Control-P zu viel.
- Die fünfte und die sechste Zeilen können zusammengefaßt werden, außerdem reicht ein <CR> : önr<RET> (in ZEX sind : ^ö = <CR><LF>, ö = <CR>, und wenn man wirklich mal ein kleines deutsches ö in einer Kommandozeile eingeben möchte, muß man dem ein Dollarzeichen davorsetzen: \$ö)
- Zeile 8 ist überflüssig, wenn ich mit ZEX in NW bin, führt der Befehl ^C nicht dazu, ZEX abzubrechen, sondern dazu, eine Seite im NW-Dokument weiterzublättern.

...und zu Info25-6, Jans Antwort :

- Mein NewWord Vers. 2.17 speichert auf die Befehle ^kx und ^kd sauber ab (^ks hab ich nicht ausprobiert).

SUB, ZEX und MENU - Wie passen sie zusammen?

(Peter Würfel, 7262)

Ich möchte hier die verschiedenen Kombinationsmöglichkeiten aufzeigen ohne Rücksicht darauf, ob diese Kombination für die Praxis relevant ist oder nicht.

1. Aufruf von der ZCPR2-Ebene (A> - Prompt)

- 1.1 Ein *.ZEX-file kann kein weiteres ZEX aufrufen
- 1.2 Ein *.ZEX-file kann SUB aufrufen. Wenn das Programm, das von diesem *.SUB-file aufgerufen wird, gestartet ist, wird das *.ZEX-file weiter abgearbeitet, sobald das aufgerufene Programm eine Eingabe via CP/M holt. Das *.ZEX-file bedient damit das Programm, das im *.SUB-file aufgerufen wurde.
- 1.3 Ein *.SUB-file kann ZEX aufrufen, alles was im *.SUB-file jedoch auf dieses ZEX folgt wird ignoriert.
- 1.4 In einem *.SUB-file können keine zwei ZEX-files aufgerufen werden.
- 1.5 Ein *.SUB-file kann weitere SUB aufrufen.
- 1.6 Am Ende eines *.SUB-files kann MENU mit Optionen aufgerufen werden. VORSICHT \$\$\$SUB ist immer noch vorhanden!

Anm.d.HzN.: Wie wäre es, wenn die letzte Zeile nicht nur MENU startet, sonder auch eben mal das dumme \$\$\$SUB löscht ?

Also statt: MENU

Die Zeile: ERA \$\$\$\$\$.SUB;MENU

Nein meine Tastatur prellt nicht, SUB verwendet das \$-Zeichen für sonderfunktionen und daher wird ein \$ als \$\$ in der .SUB-Datei angegeben. Wenn die letzte Zeile nur aus

ERA \$\$\$SUB;MENU

besteht, meckert SUB!

2. Aufruf von MENU aus:

- 2.1 In MENU kann in einer Befehlszeile nur einmal ZEX aufgerufen werden, alle weiteren Befehle in dieser Zeile werden ignoriert. Nach Beendigung des ZEX-files wird auf die MENU-Ebene zurückgekehrt.
- 2.2 Ein *.ZEX-file, das von MENU aus aufgerufen wird, kann **SUB aufrufen**
- 2.3 Ein *.SUB-file, das von MENU aus aufgerufen wird, kann **kein ZEX aufrufen**

Leserbrief: Hartmut Traber, 5270

Hartmut Traber

Hohbeuistr. 8
5270 Gummersbach

Lieber Herbert,

es ist wieder mal ein Leserbrief fällig.

1. Über den "Humor" aus Info 32 habe ich Tränen vergossen, Tränen des Lachens. So was kennt man ja eigentlich, aber zur Auflockerung finde ich das gar nicht schlecht. Ich habe mich über den veröffentlichten Humor immer amüsieren können. Soweit dieses Statement.
2. Für die Datenübertragung habe ich früher (als diese Technik für mich noch neu war, und Software rar), viel mit "Contact" gearbeitet. Anschließend kam RAMxx und "Connect" von Herbert Oppmann, machte aber nicht so viel her wie Contact. Lange Rede, kurzer Sinn:

Mit BOOTC3.COM von Michael Keßler wieder auf CP/M gegangen, mit der uralten Komm-Disk, und was tat sich:

CP/M ließ sich nicht mehr booten. D.h. es kamen zwar diverse Meldungen, die aber nicht (?) weiterhalfen (zunächst). Rücksprache mit Michael Keßler und OBh (Bootsektor) angepaßt, half nicht. Bei mir, mit folgendem Aufbau, half das Eintragen der aktuellen Laufwerkssituation von 03 bis 0B im Bootsektor. (B:, C:, D:, E: 0-Spur-Lw. mit den entsprechenden Formaten, G:, I: ist c't CMOS-Disk 256k, H: ist Memotech 1MB-SiDisk). Damit konnte ich dann "good old CP/M" und auch "Contact" fahren.

3. DIES IST KEINE KRITIK AN RAMXX.COM, SELBSTVERSTÄNDLICH AUCH NICHT AN BOOTC3.COM, GANZ IM GEGENTEIL.
4. Jede Kleinigkeit muß mitgeliefert werden und hilft Anderen!

Also:

Schreibt, was das Zeug hält!

Hartmut Traber

Anm.d.HzN.: CONNECT.COM von Herbert Oppmann ist ein Terminal-Programm welches CONTACT unter RAM x.x soweit ersetzt, wie Herbert es für sich haben wollte. D.h. CONNECT soll nicht irgendwelchen Kriterien und Ansprüchen genügen, als das tun, was Herbert für sich haben wollte. Und das tut es allemal!

Anm.d.HzN.: Ich habe in den Anfängen meiner MTXeritis auch mal versucht Datenfernübertragung mittels Akustikkoppler zu praktizieren. Mit CONTACT habe ich zum einen Probleme gehabt, fand es zum anderen auch nicht so toll. Ich habe daher seit je her das Modem-Programm MDM712 (auf CLUB.001 M1 genannt) eingesetzt, welches zusätzlich Massen-Datei-Transfer unterstützt, falls der Empfänger auch dieses Programm einsetzt. Empfänger sagt 'Massen-Datei-Empfang', anschließend sagt der Sender 'Massen-Datei-Senden' incl. der DIR-Maske wie *.DOC o.ä. und MDM712 kopiert die geforderten Dateien vom Sender zum Empfänger. Bei Übertragungsfehlern wird der defekte Block erneut übertragen. Ist alles fertig Piepts auf dem Bildschirm. Aber auch Mailbox-Zugriffe waren für mich kein Problem! Und das tolle: MDM712 läuft auch unter RAM x.x!

Software: Wo's was zu kaufen gibt

Anm.d.HzN.: Ich erhielt den u.g. Hinweis von Joachim Keiser. Die genannte Filiale in Hamburg gibt's allerdings nicht mehr.

Joachim Keiser

4925 Kalletal, 02.07.1989

Wo gibt's Turbo-Pascal zu kaufen(Info 32-4)

Für den NDR-Computer und mc-Computer wird CP/M-Software von

Graf Elektronik Systeme GmbH
Magnusstraße 13,
Postfach 16 10
8960 Kempten (Allgäu)
Tel. 0831 / 6211

vertrieben. Es gibt auch eine Filiale in 2000 Hamburg 50, Ehrenbergstraße 56, Tel.: 040 / 38 81 51.

Anfang Mai habe ich die Turbo Pascal Version 3.0 bekommen. Mit RAM 4.x kann man die 5 1/4" 80 Spur Disketten lesen (Format F6 mc-FLO-2). Die im Katalog angebotene deutsche Ausführung ist nicht mehr lieferbar (nur noch englisch). Sie kostet 309,- DM. Weitere CP/M-Software 5 1/4", 80 Spuren, laut Katalog und Preisliste vom 15.02.1989:

10 193	dBase	DM 99,-
10 442	WordStar	DM 99,-
10 331	Multiplan	DM 99,-
10 768	Haushaltsführung - ein dBase-Beispiel	DM 49,-
11 058	Microsoft Entwicklungspaket MBASIC, BASCOM, M80, L80	DM 248,-
10 522	Strukta (Pre-Compiler)	DM 349,-
11 168	SPS-Compiler	DM 298,-
11 127	Fensterverwaltung unter CP/M 2.2	DM 49,-
10 429	Turbo Pascal 3.0	DM 309,-
10 764	Turbo Tutor	DM 129,-
10 550	Disketten-Editor in Turbo Pascal	DM 39,-
10 282	HEBAS (Basic-Interpreter)	DM 97,99
11 023	BIOS Quelle	DM 49,-
10 617	Z80-CP/M-Tools	DM 139,-
10 966	BIOS für EPROM-Floppy	DM 49,-
10 724	CAD	DM 398,-
10 869	BIOS-Patch (Diskettenformate bearbeiten)	DM 95,-

Anm.d.HzN: Ich bin gerne bereit die Disketten von mc-FLO-2 auf MTX 03 zu konvertieren!

H a r d w a r e: Reparatur-Tip

Herbert Oppmann
Erwin-Rommel-Str. 59
8520 Erlangen

Erlangen, den 12.06.89

Reparatur:

Ich hatte letzte Woche den Rechner von R. Kirchhoff zur Reparatur. Es war ziemlich vertrackt, vor allem, da sich zwei Ursachen überlagerten und nicht reproduzierbare und verwirrende Symptome zeigten. Ich glaube, Ihr wolltet mal eine Reparatur-Kartei anlegen und suchtet Ursachen samt Symptomen und Abhilfen. Nun - ich kann jetzt zwei beisteuern, und zwar in der Datei REP - äh nein, REPAR ist besser.

OK, that's it for today.

Tschüß! Herbert

Hier die Datei REPAR:**Reparatur-Tips****PROBLEM:**

Rechner läuft ca. 15 min, dann Absturz (auch Reset bringt ihn dann nicht zum Laufen). Nach Ausschalten und ca. 5 min Warten geht er wieder.

URSACHE:

Der kleine Kühlwitzer auf der MTX-Hauptplatine hat 0.5 mm Abstand zu Gehäuseunterschale, anstatt plan aufzuliegen. Aufgrund mangelnder Kühlung überhitzen sich die Regler, bis die Temperaturüberwachung des 5V-Reglers (7805) anspricht. Die 5V-Versorgungsspannung sinkt ab auf ca. 2 Volt, der Spannungsabfall am 10 Ohm Widerstand steigt von kleiner 1 Volt auf 6 bis 7 Volt, wodurch sich dieser gemächlich in den benachbarten EPROM-Sockel schmilzt und stinkt.

LÖSUNG: (a)&b) Herbert Oppmann, c) HZN)

a) Kühlblech ausbauen und in Schraubstock ein wenig abkanten, damit ein guter Kontakt zur Gehäuseunterschale erzielt wird. Mit Wärmeleitpaste versehen, wieder einbauen (Glimmer-Isolierscheibe nicht vergessen!) und die Platine samt Kühlkörper durch kräftiges Zuschrauben fest nach unten ziehen.

Anm.d.HZN.: Hiervon rate ich ab, da die Platine dann unter mechanischer Spannung steht!

b) im Rahmen eines Umbaus die Spannungsregelung von der MTX-Hauptplatine runternehmen (senkt die thermische und mechanische Beanspruchung).

c) Kühlblech da lassen wo es ist, und durch das Loch des 7805, durch das bisher die Befestigungsschraube ging eine kurze Schraube setzen und auf der Oberseite mit einer Mutter festzurren. Nun sind die drei Bräter fest angeschraubt. Jetzt die Hauptplatine in das MTX-Bodenblech einlegen, d.h. dort hin wo sie hin gehört, jedoch die MTX-Plastik-Rückwand weglassen. Nun mit einem 3.5 Millimeter-Bohrer ein Loch durch die Mitte des überstehenden Teiles des Kühlblechwinzlings bohren, und dabei natürlich auch gleich durch das darunter liegende MTX-Bodenblech. Die Bohrspäne und eventuell hochstehende Reste an den Borhlöchern gründlich entfernen. Nun zur MTX-Plastik-Rückwand: Dort wo das Kühlblechle sitzt oben den Plastik-Überstand innen wegbrechen (Kneifzange). MTX zusammensetzen und die Befestigung des Kühlbleches an der Bodenplatte über das just neu geschaffene Loch vornehmen. **WICHTIG:** Da das Bodenblech hier schon unten rund ist, muß die Mutter und nicht die Schraube gedreht werden - darum wurde das Plastik auch etwas bearbeitet. Jetzt ist der Kontakt optimal und unverkantet.

H a r d w a r e: Reparatur-Tip

PROBLEM:

Egal, welches Boot-EPROM (Original oder von M.Kessler), nach Einschalten läuft Laufwerk nicht an (egal welches). Nach ca. 1 min Aus- und wieder Einschalten, und Laufwerk läuft an, geht aber bei Drücken von Reset nicht kurz aus. Uraltet Memotech-CP/M 2.2 läuft einwandfrei (wenn Laufwerk anlief), Booten von RAM x.x geht gut bis zum ClearScreen, dann hängt das System (Laufwerk läuft ewig, nur noch Reset geht).

URSACHE:

Takterzeugung auf der Floppykarte hat Knacks. Floppycontroller erhält zwar Reset, aber ohne Takt tut er gar nix (nicht mal HeadLoad ausmachen, welches bei uns für MotorOn verwendet wird). Im 5"-Modus (1 MHz Takt) ging der Rechner nach einer einminütigen Warmlaufphase, im 8"-Modus (2 MHz Takt) ging gar nix. IC E4 (74LS157) und/oder IC E3 (74LS74) hatten eine Macke (ich habe beide gleichzeitig zerstörend ausgebaut, deshalb kann ich nicht mehr sagen, welcher von beiden nun genau der Übeltäter war).

LÖSUNG:

IC E4 und/oder IC E3 auf der Floppykarte austauschen.

Anm.d.HzN.: Wenn Du viel Glück hast, ist IC E5 (72LS74) auch noch fällig!

A propos kaputte IC's beim Floppy-Controller: (Herbert zur Nedden, 2000)

PROBLEM:

Laufwerk scheint richtig anzulaufen, meldet gelegentlich auch noch, daß es booten will - aber dann tote Hose. Schiebe ich den Kopf jedoch auf eine innere Spur tut sich nichts, d.h. der Kopf bleibt wo er ist.

URSACHE:

Verutlich kommt kein STEP-Impuls beim Laufwerk an. Die Ursache ist hier wahrscheinlich das IC E1 (74LS123), ein Mono-Flop (im Gegensatz zum Mega-Flop, denn da bedeutet Flop Floating Point Instruction, während Mono-Flop wohl mehr damit zu tun hat, daß immer nur EINE Floppy angesprochen wird, oder ???). Spaß beiseite, Ernst komm Du mal her. Also der 74LS123 liefert den STEP-Impuls, und kann auch mal klein beigegeben.

LÖSUNG:

IC E1 auf der Floppykarte austauschen.

Hardware: Tastatur-Zeichensatz ?**US-Zeichensatz für die Tastatur**

(Herbert zur Nedden, 2000)

Frage: (eines Clubmitgliedes)

Ist es möglich, unter RAM4 auf die englische Tastatur umzuschalten ?
(C-Programme, in denen relativ oft die eckigen und geschweiften Klammern vorkommen, sind mit der deutschen Tastatur recht unleserlich).

Antwort:

Es ist nicht eine Frage der Tastatur, ob auf dem Bildschirm ein Ä oder die große eckige Klammer erscheint, da beide unter CP/M den selben Code haben, also über die selbe Taste eingegeben werden. Was auf dem Bildschirm sichtbar ist, ist eine Frage des Zeichensatzes der 80-Zeichen-Karte.

Den Zeichensatz der 80-Zeichen-Karte kann man umschalten mit

Esc S Standard-Zeichensatz mit Umlauten
Esc A Alternativ-Zeichensatz ohne Umlaute
Esc G Grafik-Zeichensatz

Der Original-Memotech-Alternativ-Zeichensatz hat an Stelle der Umlaute des Standard-Zeichensatzes Dinge wie bestimmte Brüche und Pfeile (soweit ich erinnere). Mit dem Alternativ-Zeichensatz aus meinem 80-Zeichen-EPROM erscheinen an Stelle der Umlaute die geschweiften und eckigen Klammern.

Wie gesagt, im Gegensatz zu MsDos kennt CP/M nur einen Code für Umlaute und die eckigen Klammern (Ä = [, ä = {, ...). Je nachdem, was ich meiner 80-Zeichen-Karte nun verklickere stellt sie Umlaute als solche dar, oder auch nicht. Steckt mein 80-Zeichen-EPROM auf eben dieser Karte erscheinen an Stelle der Umlaute halt die ersehnten Klammern.

Um den Zeichensatz umzuschalten genügt es, Esc A beim Laden von Turbo-Pascal oder C auf den Bildschirm zu senden, was i.a. mittels des installierbaren sogenannten Terminal-Initialisierungs-String machbar ist. Nach dem Verlassen von Turbo-Pascal oder C wird (Terminal-UnInitialisierungs-String) Esc S abgesetzt, damit ich wieder Umlaute erhalte, so ich das will.

Hinweis:

Damit der Drucker nun auch die Klammern liefert, muß diesem das durch Umschalten auf den US-Zeichensatz mitgeteilt werden. Der DMX 80 tut dies wie die meisten EPSON-Kompatiblen Drucker auf Esc R ^§ (^§ = Hex-Null) hin.

Bemerkung:

Mein Alternate-Zeichensatz der 80-Zeichen-Karte ist nichts anderes als der US-Zeichensatz, während der Standard-Zeichensatz der deutsche ist.

Übrigens:

Olaf Krumnow's KLIX-Hardcopy ist in der Lage zu erkennen, ob auf der Glotze der Standard- oder Alternativ-Zeichensatz aktiv ist, und druckt den Standard als deutschen, den Alternativ als US-Zeichensatz, so das gewünscht wird.

Hardware: Port-Adressen**Port-Adressen**

(Herbert zur Nedden, 2000)

Vor Ewigkeiten dachte ich daran, daß es evtl. Sinn machen könnte, die Port-Adressen diverser ECB-Karten umzulegen. Mittlerweile bin ich davon abgekommen. U.a. müßten Boot-EPROMS wegen der Conitec-DISI geändert werden und, und, ..

Damit und mit den Neuerungen sind die Adreß-Belegungen wie folgt:

- In **fett** die Adressen, die **NICHT** geändert werden sollten, d.h. die, die bei RAM 5.x zu Grunde gelegt werden!
- Für Hinweise auf Fehler bin ich dankbar.

Bereich	Port	Verwendung
MTX	00	Paging-Port und Centronics-Strobe
	01 - 02	VS4-Controller
	03	Ton, Cassette, Alpha-Lock-Lämpchen
	04	Centronics (Druckerport)
	05 - 06	Tastatur und Ton
	07	Port 7, i.a. Hardware-Uhr
	08 - 0B	CTC
	0C - 0F	DART (RS232)
	10 - 14	SDX-Floppycontroller
	15 - 17	frei (3)
	18 - 1B	8255-PIO-Box oder 2. CTC
	1C - 1F	2. DART, 1F = Cassette
	FDX	20 - 2F
30 - 33		80-Zeichenkarte: RAM's
34 - 37		frei (4)
38 - 3B		80-Zeichenkarte: 6845 + HGR
3C - 3D		frei (2)
3E		Hardware-Uhr statt auf Port 7
3F		frei (1)
40 - 47		FDX-Floppycontroller
48 - 4F		HardDisc-Controller
50 - 5F		Memotech-SiDisc
60 - 7F		frei (32)
ECB	80 - 83	Edicta-Grafik
	84 - 8F	frei (12)
	90 - 97	8255-PIOs
	98 - 9B	c't RAM-Floppy
	9C - 9F	frei (4)
	A0 - A3	Edicta RAM-Floppy
	A4 - A7	c't EPROM-Floppy normal
	A8 - AB	c't SRAM-Floppy
	AC - B7	frei (12)
	B8 - BB	Conitec DISI SRAM-Floppy normal
	BC - BF	Conitec DISI SRAM-Floppy zum Booten
	C0 - CB	frei (12)
	CC - CF	Janich & Klass Programmer
	D0 - DF	I/O-Karte (c't CTC, SIO, 2xPIO o. J&K 4xPIO)
	E0 - E7	frei (8)
	E8 - EF	HardDisc-Controller
	F0 - F7	frei (8)
	F8 - FB	HD 64180 Sub-Prozessorkarte
	FC - FF	frei (4)

H a r d w a r e: FDX-OMTI-Adapter / S o f t w a r e: KCLICK.007

FDX-Adapter-Einschubkarte für ECB und OMTI

(Herbert zur Nedden, 2000)

Gerhard Witzel hat ja, wie in Info 32 erwähnt, einen FDX-Einschubkarte mit einem ECB-Steckplatz und Platz für eine Hardware-Uhr entwickelt, die es bei ihm zu kaufen gibt. Nun hat Claudio Romanazzi aufgezeigt, wie der IBM-Harddisc-Controller OMTI an den ECB-Bus, und damit an dem Memotech angeschlossen werden kann. Will man also die HardDisc in die FDX stecken, kann man ja Gerhard's FDX-ECB-Adapter verwenden, und dort hinein einen ECB-OMTI-Adapter stecken, der den OMTI aufnimmt.

Da mir hierfür der relativ aufwendige FDX-ECB-Adapter zu schade schien (der OMTI benötigt lediglich die Signale D0-D7, A0-A7, IOREQ, RD, WR und M1, z.T. über ein paar Gatter (2 IC's) geführt, um an Adresse 48h betrieben zu werden), lag es für mich nahe, einen FDX-OMTI-Adapter zu entwickeln, der obendrein den Einbau erheblich einfacher gestalten würde.

Diese Idee habe ich, unterstützt durch Claudio, Gerhard Witzel unterbreitet, der sie in einen FDX-OMTI-Adapter incl. Platz für eine Hardware-Uhr und AD-Wandler (Claudio's Wunsch) umsetzte.

Hinweis: Firma OMTI darf dank einer erfolgreichen Klage der IBM ihre bekannten OMTI-Controller nicht mehr herstellen, so daß es sicherlich nicht ganz einfach ist, diese Teile zu erwerben.

Firma Edicta, Löwenstr. 68, 7000 Stuttgart 70, 0711 - 763381 bietet die OMTI's noch an. Der 5510 kostet z.B. DM 190.-

Die James-Bond-KLICK: KCLICK.007

(Herbert zur Nedden, 2000)

Auf KCLICK.007 ist natürlich Claudio's HardDisc-Treiber.

Weiterhin hat Michael Keßler seine FORMSTAR-Sammlung als PD erklärt, die ebenfalls anbei ist. INITDIR hinterlegt nun auf Disketten ohne Systemspuren ein Format-ID!, so daß auch diese Formate automatisch sicher erkannt werden können. CFG5.KLX, welches nur FDX-tauglich ist, um einen BDOS-Diskreset erweitert, der die im letzten Info genannten Probleme behebt, und bislang bei Michael keine Probleme hervorrief - er hatte Glück, daß das BDOS immer da war! Für die FDX hat er MFT52 um eine automatische Format-Erkennung erweitert, und bei dieser Gelegenheit die Bildschirmausgabe gefENSTERT (wem's gefällt!) und eingedeutscht. Da er schon einmal dabei war: Diverse Utilites haben einen neuen Bildschirmaufbau, der bei DDIR einen Ausdruck via ^P unmöglich macht, aber dafür gibt's ja Hardcopy im KCLICK.

Lang ersehnt und endlich da: Olaf's Dijey in einer BDOS-Error-abfangenden Version. Natürlich gleich mit neuen Kommandos wie u.a. das lapidare Vergleichen von Dateien oder das Mal-Eben-Erstellen einer Datei, in der alle *.MAC-Dateinamen gleich mit dem Aufruf des Assemblers drinne stehen. Die Kurzhilfe wurde um ein Help-Overlay DJHelp.KLX ergänzt. Will ich unter DiJey Help (dazu drücke ich auf das Fragebein ?) erscheint die Kurzhilfe, falls DJHelp nicht geladen ist. Ist DJHelp jedoch auch im KLIX-Heap wird dieses aufgerufen. Oft nach gefragt - endlich PD ist der Cache der Diskettenzugriffe beschleunigt - siehe auch letztes Info.

Was sonst noch so auf KCLICK.007 kommt wird sich zeigen - ich weiß nicht, wieviel Platz Claudios HardDisc-Treiber belegt.

S o f t w a r e: COMPARE

C O M P A R E --> CLUB.045

(Herbert zur Nedden, 2000)

COMPARE vergleicht zwei Text-Dateien und meldet zweierlei Informationen:

1. Zeilen bzw. Zeilenpassagen, die sich verändert haben.
2. Zeilen bzw. Zeilenpassagen, die eingefügt bzw. gelöscht wurden

COMPARE merkt es auch, wenn in einer Datei 30 Zeilen eingefügt, bzw. aus ihr gelöscht wurden. Ich glaube bei 150 (oder 120 (oder so)) ist die Grenze erreicht. Allerdings werden dicht beieinanderliegende Zeilenänderungen plus Einfügungen/Löschungen i.a. gesamt als Änderungen angegeben, was was daran liegt, daß die Synchronisation, d.h. der Glaube von COMPARE, daß es wieder auf gleiche Zeilen gestoßen ist nicht anhand von einer, sondern von mehreren Zeilen erfolgt. Man denke nur an ein Assembler-Proramm, in dem einige Zeilenfolgen immer wieder auftauchen. Immerhin klappt die Synchronisation fabelhaft.

Hier die Beispiel-Dateien:

HEX1.MAC:

```

;-----
; HexByte in Speicher im Display-Format ablegen
;-----
HexToMem:    push    af
             srl     a
             srl     a
             srl     a
             srl     a
             call   HeToMeNib
             pop     af
             and     0fh
HeToMeNib:   add     a,90h
             daa
             adc     a,40h
             daa
             ld     (h1),a
             inc    h1
             ret

```

HEX2.MAC:

```

;-----
; HexNibble in Speicher im Display-Format ablegen
;-----
HexToMem:    push    af
             srl     a
             srl     a
             srl     a
             srl     a
             call   HeToMeNib
             pop     af
             and     0fh
;-----
; HexNibble in Speicher im Display-Format ablegen
;-----
HeToMeNib:   add     a,90h
             daa
             adc     a,40h
             daa
             ld     (h1),a
             inc    h1
             ret

```

S o f t w a r e: COMPARE

HEX3.MAC:

```

;-----
; HexByte in Speicher im Display-Format ablegen
;-----
HexToMem:      push    af
               srl     a
               srl     a
               srl     a
               srl     a
               call   HeToMeNib
               pop     af
               and     00001111b
HeToMeNib:    add     a,90h
               daa
               adc     a,40h
               daa
               ld     (h1),a
               inc    h1
               ret
    
```

COMPARE von HEX1.MAC und HEX2.MAC:

```

A>compare
INPUT FILE NAME:
hex1.mac
SECOND INPUT FILE NAME:
hex2.mac
    
```

Source Compare [v203 5-3-82]

```

*****
EXTRA TEXT ON hex2.mac, LINES 12 TO 14
BETWEEN LINES 11 AND 12 OF hex1.mac
    
```

```

* ;-----
* ; HexNibble in Speicher im Display-Format ablegen
* ;-----
    
```

COMPARE von HEX1.MAC und HEX3.MAC:

```

A>compare
INPUT FILE NAME:
hex1.mac
SECOND INPUT FILE NAME:
hex3.mac
    
```

Source Compare [v203 5-3-82]

```

*****
MISMATCH:
    
```

```

hex1.mac, LINE 11:
*           and     0fh

hex3.mac, LINE 11:
*           and     00001111b
    
```

S o f t w a r e: BACKitUP

Programmbesprechung BACKitUP (von Holger Hansen)

(Olaf Krumnow, 2050)

BACKitUP ist der Nachfolger des bereits seit längerem erhältlichen Programms BACKUP. Es hat einige Erweiterungen erfahren, so daß ich mittlerweile doch auch neugierig wurde, was sich hinter diesem Programm denn nun genau verbirgt.

BACKitUP soll helfen, Sicherheitskopien von Disketten zu unterstützen und zu beschleunigen, indem nur die Dateien, die auch wirklich verändert wurden, kopiert werden. Dazu orientiert sich BACKitUP an den Zeiteinträgen, die P2DOS im Directory anlegt. Daraus folgt, daß die Disketten für Zeiteinträge initialisiert sein müssen. Beim Kopieren wird der Zeiteintrag der Quelldatei für die Zieldatei übernommen, was eine wichtige Voraussetzung für ein sinnvolles Arbeiten ist. Leider wird bei der Zieldatei als Create-Zeit immer die Update-Zeit der Quelldatei übernommen. Besser fände ich es, wenn die Create-Zeit des Originals (also der alten Zieldatei) erhalten bleiben würden. Dann hätte man bei konsequenter Benutzung von BACKitUP stets einen Überblick über die ursprünglichen Zeiten.

Ich persönlich benutze BACKitUP dazu, beim Programmieren, das bei mir grundsätzlich auf der RAM-Floppy stattfindet, die Arbeit auf eine 'feste' Diskette zu sichern. Bisher hatte ich die Möglichkeit, anhand der vorhandenen BAK-Dateien festzustellen, welche Dateien mittlerweile geändert wurden, und die entsprechenden Original-Dateien von Hand zu selektieren und zu kopieren. Oder aber ich mußte der Sicherheit wegen alle Dateien kopieren. Mit BACKitUP nun geht das alles wesentlich einfacher und schneller, denn es werden nur die veränderten Dateien kopiert. Netterweise unterdrückt BACKitUP auch noch das Kopieren von BAK-Dateien, so daß ein vorheriges Löschen dieser lästigen Dateien unterbleiben kann. Auch für regelmäßige Sicherheitskopien von 'festen' Disketten, die man ja doch immer mal machen sollte, ist BACKitUP ideal. Gerade bei den großen Disketten verliert man oft den Überblick. Und wenn nur wenige Dateien geändert wurden, so ist BACKitUP ganz sicher die schnellste Methode. Dabei ist weiterhin von Vorteil, daß BACKitUP auch User unterstützt. Es können sogar alle Userbereiche auf einmal gesichert werden. Ein enormer Gewinn, gab es doch bisher kein Kommandozeilen-orientiertes Programm, das dieses kann.

Positiv fiel weiterhin die hohe Kopiergeschwindigkeit auf, die durchaus mit NSWEEP oder DiJey mithalten kann. Das liegt sicher auch daran, daß der gesamte noch zur Verfügung stehende Speicher als Kopierpuffer benutzt und sogar dynamisch vergrößert wird, in dem Maße, wie Dateien abgearbeitet und der für das Directory benötigte Platz freigegeben wurde. Da BACKitUP auch über einen reinen Kopiermodus verfügt (es ist dann noch etwas schneller, da der Zeitvergleich mit der Zieldatei entfällt), kann es auch gut als universelles Kopierprogramm benutzt werden, besonders da es über die Kommandozeile gesteuert werden kann, was bei NSWEEP oder DiJey ja nicht möglich ist.

BACKitUP bietet die Option, in der Kommandozeile eine Zeit anzugeben. Dann soll es nur die Dateien kopieren, die neuer sind als angegeben. Leider muß da irgendwo ein Bug in der Auswertung der Kommandozeile sein, denn immer wenn ich Datum und Zeit mit eingab, fing BACKitUP an, mich genau danach zu fragen. Wenn ich die Daten dann eingegeben hatte, kopierte BACKitUP munter alle gefundenen Dateien, unabhängig vom Datum. Andererseits ist diese Option aber auch nicht so wichtig, jedenfalls bei mir nicht.

BACKitUP soll auch BackUps auf kleinere Disketten unterstützen, also z.B. Dateien von einer 1.6MB-Diskette auf mehrere 390kB-Disketten verteilen. Leider klappte das bei mir nicht, da sich das Programm nach dem Diskettenwechsel verabschiedete und nur noch via KLICK zu verlassen war. Aber selbst dann müßte BACKitUP vor dem Kopieren erst die Directories aller Zieldisketten lesen, um Änderungen korrekt festzustellen. Und das tut es bisher nicht. Auch würde mich interessieren, ob BACKitUP es unterstützt, große Dateien, die nicht auf die Zieldiskette

Software: Neues in der PD-Welt

passen, zu splitten und bei Bedarf wieder richtig zusammzusetzen. Das scheint mir insbesondere in Hinsicht auf die zunehmende Harddisk-Aktivität im Club ein nicht ganz unbedeutender Aspekt zu sein. Ferner fiel störend auf, daß sich BACKitUP via BDOS-Error verabschiedete, wenn es eine R/O-Datei kopieren sollte. Da auch Public-Files R/O sind, kann das Backups doch gewaltig stören.

BACKitUP patcht das P2DOS, um dafür zu sorgen, daß die ursprüngliche und nicht die aktuelle Zeit in das Directory eingetragen wird. Da ich nun aber bereits das geänderte P2DOS von RAM4.5 benutze, Holger aber noch nicht, stürzte bei mir zunächst ständig das Programm ab. Erst nach Änderung im Source-Code und Neuübersetzung klappte das. Warum wurde da nicht der BIOS-Einsprung verändert (den kann man immer lokalisieren!)? Mit einem kleinen Flag kann doch sichergestellt werden, daß wirklich nur bei Bedarf die alte Zeit geliefert wird.

Die genannten Probleme sind vielleicht auch auf fehlerhafte Bedienung zurückzuführen. Da mir aber außer des kurzen Hilfstextes, den BACKitUP selber liefert, keine Dokumentation zur Verfügung stand, kann ich das nicht beurteilen.

Das Fazit ist kurz: BACKitUP ist bei mir im regelmäßigen, zufriedenstellendem Einsatz und das sagt alles!

Anm.d.HzN.: Holger hat zugesagt, daß er sich der o.g Fehlern annehmen will, und bei dieser Gelegenheit die eine oder andere Kleinigkeit an BackItUp bereinigen wird. Das wird evtl. etwas dauern, da er sich zur Zeit mit Prüfungen herumschlagen darf. Anschließend wandert BackItUp auf jedenfall auf CLUB.045 als Public-Domain!

Neue PD-Software

(Herbert zur Nedden, 2000)

Da ich nun meine erste Turbo-Pascal-Anwendung, die mit Pointern arbeitet fertiggestellt habe, gibt es die SYSLIB Version 3.6 sowohl mit 6-Zeichen-Labels im Microsoft-Format (M80/L80 und Konsorten) als natürlich auch im SLR-Format mit 16-Zeichen-Labels auf CLUB.043. Mein Programm LIBCNTL analysiert nämlich die Z80-Assembler-Sources, ermittelt die EXTERNALs und PUBLICs und liefert u.a. die Liste der Dateien in der Reihenfolge, wie sie in die Library, in diesem Fall die SYSLIB, mittels des Librarian (LIB80 bzw. SLRIB) eingestellt werden sollten. Die Reihenfolge ist insofern von Bedeutung, als der Linker (L80 bzw. SLRNK) die Libraries nur einmal durchforsten, und daher interne EXTERNAL-PUBLIC-Paare in eben dieser Reihenfolge auftreten müssen, damit der Linker erst einmal erfährt, daß der Label benötigt wird, bevor er dessen Definition via PUBLIC findet. Trifft der Linker nämlich auf einen PUBLIC, der noch nicht via EXTERNAL angefordert wurde, läßt er ihn weg - was ja just die Idee des Linkers ist!

Auf CLUB.044 landet zusammen mit LIBCNTL Herbert Oppmanns ReadTrack, wieder einmal DiscDoc (war 'n Fehler drin), ein Patch für Turbo-Pascal, MacForm zum formatieren von Assembler-Sources sowie Programme für den Betrieb des Selbstbau-Plotters MONDRIAN aus Elektor 1/88.

CLUB.045 wird außer BackItUp diverse Programme der MT-Pascal Users-Group enthalten, wie u.a. das o.g. COMPARE. Da hier noch etwas Platz drauf ist kommt noch etwas dazu - aber keiner weiß was!

Im SIG/M-Bereich gibt's wieder ein paar neue Scheiben: Die zweite Hälfte meiner hier gestarteten Software-Tausch-Aktion. Näheres: siehe Angebotsliste.

R A M 5.x: Planung**Was ist in Sachen RAM 5.x geplant**

(Herbert zur Nedden, 2000)

WICHTIG: NEIN, RAM 5.x ist noch nicht fertig und daher auch noch nicht zu haben. Olaf Krumnow und ich haben erst einige Teile teilweise programmiert. Es wird auch noch etwas dauern - wir hoffen auf Weihnachten!
 Hier will ich Euch lediglich schon einmal den Mund wässrig ... äh nein Euch wissen lassen, was wir so zum Teil vor haben, um Euch die Gelegenheit zu geben, Eure Ideen und Wünsche zu äußern.

Grundsätzliches:

RAM 5.x wird samt ZCPR3 als Komplettsystem mit einem neuen Handbuch ausgeliefert. D.h. alles was RAM 4.x/ZCPR2/P2DOS heißt kann weg - es wird ersetzt. Natürlich werden wir eine 'Schwarze Liste' von Dateien aufstellen, in der die Dinge zu finden sind, die auf jeden Fall weg müssen, da sie unter RAM 5.x garantiert Mist machen oder lediglich durch Besseres ersetzt werden.
 Wir haben vor, die Installation menügeführt anzubieten, d.h. Diskette einlegen, Memotech einschalten und den Anweisungen auf dem Bildschirm Folge leisten. Damit RAM 5.x nicht unnötig umfangreich wird, werden die Hilfsprogramme, die kein Muß sind, auf PD-Scheiben angeboten. Wer die ZCPR3- und P2DOS-Doku aufbereiten und/oder gar eindeutschen mag, bitte (040) 200 87 04 anrufen.

Laufwerks-Treiber:

1. Laufwerke A: bis P: einschließlich werden unterstützt, wobei wie gehabt A: und F: identisch sein werden (vom BIOS her sogar identischer als jetzt!), Disketten nur auf B:-E: und Festplatten nur auf J:-P: zugelassen werden. Die acht Config-Codes aus dem Common (Offc8h, ...) sind dann nicht mehr dort, da da kein Platz für A:-P: ist.
2. Die Formatbeschreibung im XDpb, d.h. dem, was in FORMATE.DAT steht wird erweitert: Die Anzahl von Spuren wird in zwei Bytes abgelegt, der Typ des Formates und zugehörigen Laufwerkes (Eigene Routine, Floppy, Festplatte, welche RAM-Floppy) sowie die Port-Adresse für den Zugriff kommt hinzu. Wer nun zwei identische RAM-Floppies betreiben will, muß lediglich zwei Formate definieren, die sich im Port unterscheiden.
3. Der Systemeinsprung CONFIG (Offf0h) wird INKOMPATIBEL erweitert. Er übernimmt nun auch die Formaterkennung, kann auskonfigurieren und verwaltet die CP/M-Tabellen incl. Allocation- und Checkvector. Damit reduziert sich CFGX.COM auf die Auswertung der Kommandozeile. Das KLICK-Menü erhält ein CONFIG-Menü, über welches die Laufwerke (um)configuriert werden können (CFGX.KLX entfällt). FORMATE4.COM wird durch ein 'großes' CONFIG.COM abgelöst, welches einen wesentlich bequemeren Format-Editor enthält, und automatisch beim Configurieren das gewünschte Format lädt, so es noch nicht im System ist.
4. Da der Platz zwischen FREE und TOAM rar und kostbar ist, jedoch CP/M dort die Allocation-Vektoren etc. erwartet, sowie den DPB und DPH im Common benötigt, werden diese Tabellen von DriveSelect gepaged, d.h. nur die Tabellen des aktuellen Laufwerkes sind auf Bank 0. Die anderen stehen auf dem Heap.
5. Nun liefern auch Zugriffs-Fehler auf RAM-Floppies das Fehlerfenster, welches von Laufwerks-Zugriffsfehlern her bekannt ist. Hast Du also vergessen den Schreibschutz zu entfernen, macht das nichts.
6. Olaf Krumnow's Cache wird fest in den Treiber integriert.

Funktionstasten:

Die Funktionstastentabelle, die im Common, d.h. auf Bank 0 liegt, und eigentlich überflüssig ist, verschwindet von dort. Das hat den angenehmen Vorteil, daß beim Umschalten der Funktionstastentabelle die aktuelle Funktionstaste nicht abgeborochen wird. Ich kann also auf eine Taste problemlos den Aufruf von Turbo-Pascal incl. der Einstellung der Optionen 'COM' und 'End-Adresse = 0A000h' legen: 'turbo^MOCEA000^MQ'.

RAM 5.x: PlanungUhren-Interrupt:

1. Der Uhren-Interrupt wird Dank einer Anregung von Herbert Oppmann und einiger Überlegungen meinerseits drastisch in seiner Verarbeitungszeit verkürzt, und so der Rechner schneller.
2. Für User-Interrupt-Routinen wird auf Bank 1 etwas Platz (vermutlich schlappe 256 Bytes) reserviert, so daß diese nicht mehr unter TOAM abgelegt werden müssen. Beim Aufruf der User-Interrupt-Routine ist dann auch Bank 1 aktiv.

RAM-Jumps:

1. BnkLdir wird einen schnellen BankMove zwischen einer beliebigen Bank und dem Common bereitstellen. Dabei müssen in BC, DE und HL die Transport-Informationen wie für LDIR gesetzt und im Akku die Bank übergeben werden. BnkLdir ist immerhin über dreimal so schnell wie ein BankMove, kann aber Daten nur zwischen einer beliebigen Bank und dieser Bank selbst transferieren. (Ich brauchte diese Routine für das Paging im Laufwerkstreiber und wollte ihn Euch nicht vorenthalten.)
2. Utility wird diverse Hilfsroutinen bereitstellen, die RAM 5.x-Interna verwenden. U.a. sind vorgesehen:
 - Zugriff auf die Funktionstasten wie Tabelle umschalten, zurückschalten, Tastenbelegung abfragen und setzen.
 - Zugriff auf den Tastaturpuffer wie Zeichen einstellen, Puffer löschen, fragen, was das nächste Zeichen ist.
 - Uhren und Datumsroutinen wie Uhrzeit setzen, Datum zwischen CP/M-5-Byte-Format und Klarschrift hin/her-Konvertieren.
 - Aufruf des von Laufwerks-Zugriffsfehlern her bekannten Blinke-Fensters, selbstverständlich mit der Möglichkeit eigene Texte darin auszugeben.
 Da Olaf und ich die Weisheit nicht mit Löffeln (geschweige denn mit Kellen) gefressen haben, also hier sicherlich die eine oder andere Funktion vergessen werden, ist dieser Einsprung über ein spezielles KLICK-Overlay erweiterbar.

Spooler:

1. Die Spooler-Zugriffsfunktionen im KLICK werden erweitert. U.a. wird es möglich sein, mehrere Strings zu definieren, die von hier aus an den Drucker geschickt werden können. Außerdem kann die Größe des Spoolers verändert werden.
2. Von den Spooler-Blinkefenstern 'Timeout' und 'Seitenwechsel' kann man in die Spoolerverwaltung springen - und z.B. den Puffer löschen, Spooler auf Pause setzen Einzel-Endlos-Verarbeitung umschalten, ...

Aufruf von RAM 5.x:

Der Aufruf von RAM 5.x wird weiterhin die Möglichkeit bieten den für die RAM-Floppy und den Spooler zu vergebenen Platz anzugeben. Auch die /NF-Option, die das Formatieren von F: unterdrückt wird bleiben. Zusätzlich wird es die /NK-Option geben, die das Löschen der Programme aus dem KLICK-Heap unterdrückt. Die F-Option zum Laden einer .KBD-Datei wird allerdings entfallen - das übernimmt die Installation.

ZCPR3:

Holger Hansen hat sich bereit erklärt, einen Artikel zu schreiben, in dem er aufzeigt, was das feine des ZCPR3 gegenüber dem ZCPR2 ist. Da er zur Zeit mit Prüfungen beschäftigt ist, folgt dieser Artikel im nächsten Info.