

MTX *User-Club Deutschland*

Info 34
Freitag, der 13.10.1989

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:	
Fehlerbeschreibungen	Seite 2
Korrektur & Nachtrag	Seite 3
Turbo-Tip	Seite 3
Fragen & Antworten	Seite 4
S o f t w a r e:	
MTX-Edit	Seite 3
L e s e r b r i e f:	
Claudio Romanazzi, 3070	Seite 5
Jan Brederke, 2000	Seite 9
Hans Gras, NL 1506	Seite 11
R A M 5.x:	
ZCPR3	Seite 11
S u p e r C a l c:	
Rangliste mit SC erstellen	Seite 16
C P / M:	
XDIR	Seite 20
A s s e m b l e r:	
Tips	Seite 21
Macros	Seite 25
H a r d w a r e:	
c't-SRAM-Floppy	Seite 29
80-Zeichenkarte und höhere Geschwindigkeit	Seite 30
80-Zeichenkarte und Grafik	Seite 31
Abfrage, ob Diskette gewechselt wurde	Seite 36
K o m i k:	
'n Kreuzwort-Rätsel	Seite 7
Noch'n Gedicht	Seite 28

Preis für dieses Info: DM *10.50*

Redaktionsschluß für's nächste Info: 22.11.89
(Was jedoch früher fertig ist, bitte auch früher abschicken!)

Quelle für OMTI Festplatten-Controller

- Sysline Computer, Dieselstr. 4, 8044 Lohof, 089 - 310996 und 089 - 317199
- bsc-GmbH, München liefert OMTIs XT-HDC-Modell für MFM oder RLL für ca. 200.-

Clubtreffen

(Herbert zur Nedden, 2000)

Geplant für April 1990 mit Übernachtung im Großraum Hannover!

Liebe Leserin, lieber Leser,

Es ist erstaunlich, wie schnell die Zeit vergeht. Wie letztes Jahr habe ich vor, im April 1990 ein Clubtreffen ins Leben zu rufen. Das soll dan möglichst so ähnlich, wie letztes Jahr mit Übernachtung in einem Gasthof vor Ort ablaufen. Bitte denk schon mal darüber nach, und laß mich wissen, ob Du irgendwelche Einschränkungen oder Präferenzen in Sachen Termin hast.

Olaf hat zugesagt, eine Artikelserie zu starten, in der er aufzeigt, wie ein Klick-Programm erstellt werden kann, was dazu erforderlich ist, wie das eigentlich und überhaupt funktioniert usw.

Übrigens ist einem Handbuch zu WordStar Release 4 zu entnehmen, daß MicroPro in der Tat NewStar aufgekauft hat und WordStar Release 4 ein Upgrade von NewWord ist.

Wie bringe ich meine Memotech-Programme unter MsDos zum laufen? Die klassische Variante 'man lade einen Emulgator ...' ist i.a. nervtötend. Es geht auch anders! Hartmut Traber hat ein Konverter-Programm, mit dem ein CP/M.COM in ein MsDos.COM umgesetzt werden kann. Erst unter CP/M das Programm für MsDos zu installieren, konvertieren und dann unter MsDos aufrufen. Dieses Konverter-Programm unterstützt auch die Verarbeitung selbst-definierter Disketten-Formate wie 09, 0A, 1B. LEIDER: Die Programme, die mehr als CP/M verlangen, d.h. den Memotech, ggf. mit RAM x.x benötigen können nicht auf MsDos portiert werden. Wer Interesse hat schicke eine auf MsDos formatierte Diskette samt frankiertem Rückumschlag an Hartmut Traber. Ich weiß zwar nicht wozu das gut sein soll, aber das ist nicht mein Problem!

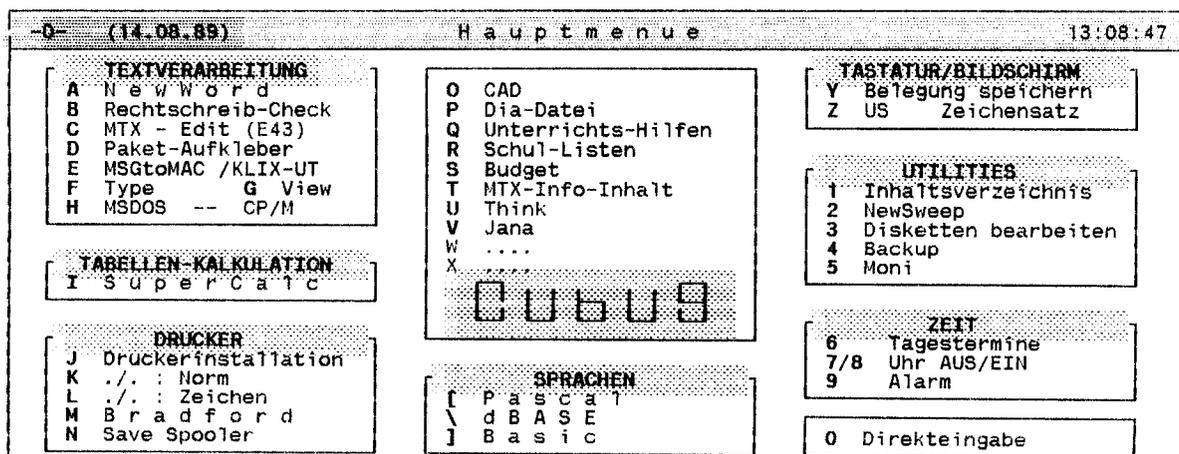
Holger Hansen, (3300) plant eine Fullscreen Shell mit Macros und Mausunterstützung. Er sucht noch Anregungen, Tips und Hinweise

Übrigens habe ich schon zum zweiten Mal eine 5 1/4"-Diskette im Format 1C erhalten. Um diese Lesen zu können muß bekanntlich das Laufwerk ungearbeitet werden, damit es im HighDensity-Modus mit 300 statt 360 Umdrehungen läuft. Ich habe von derartigen Dingen Abstand genommen! Folglich konnte ich diese 5 1/4"-1C-Disketten nicht lesen! Also bitte bei 5 1/4"-Disketten max. Format 1B verwenden, es sei denn, ich soll eh nicht draufschauen.

Als das Info druckfertig war wog ich mal nach, was denn so an Proto fällig würde. 255 Gramm! Also kosten die letzten 5 Gramm DM -.60 Porto. Nein, das war nun wirklich nicht einzusehen! Daß die Druckerei ca. 10% Preiserhöhung fordert müssen wir leider verknusen, aber die 5 Gramm Übergewicht mußten weg. Und es hat geklappt. Die Seiten 2-4 der Angebotsliste (PD-Übersicht) sind diesmal nicht dabei. Die Neuerungen dieser Aufstellung sind auf Seite 1 der Angebotsliste zu finden.

Euer Herbert zur Nedden

Wozu brauchen wir Microsoft-Windows. Peter Würfel's Memotech meldet sich nach dem Booten mit folgender Oberfläche, die Dank MENU.COM unter RAM 4.x läuft. Per Tastendruck wird die jeweilige Funktion aufgerufen - oder es erscheint ein entsprechendes weiteres Auswahlmenü ähnlicher Qualität.



Club: Fehlerbeschreibungen**Wie sag ich's dem, der den Fehler beseitigen soll** (Herbert zur Nedden, 2000)

Manchmal ist es gar nicht so einfach, wenn es darum geht, einen Fehler zu beseitigen - schließlich will er erst einmal gefunden werden.

Bei der Suche nach einem Fehler ist es i.a. hilfreich, nicht nur das Teil in die Hand zu bekommen, wo der, den der Fehler stört, eben diesen vermutet! Nicht allzu selten liegt die Fehlerursache ganz woanders.

- Stell Dir vor, Deine FDX will nicht booten, Dein Grundgerät tut jedoch anstandslos unter ROM-Basic. Klar, FDX eintüten und bei Herbert reparieren lassen. Pech ist nur - und das kommt vor -, wenn der Fehler im Grundgerät zu finden ist!
- Krasser ist vielleicht der Fall (und der ist echt vorgekommen), daß die FDX kein Lebenszeichen auf dem Bildschirm von sich gibt, und auch der Reservermonitor nichts liefert. Wer kommt auch auf die Idee, daß das Monitor-Kabel schuld ist ?
- Oder Peter Würfel mit seinen Problemen, NewWord und ZEX unter einen Hut zu bekommen. Bei Ihm tat es nicht, bei mir ohne Probleme. Schuld war weder ZEX noch RAM, geschweige denn seine .ZEX-Datei. Nein die Installation von NewWord machte NewWord zu speicherhungrig, da Peter den Unerase-Buffer mit 5000 Bytes reichlich groß wählte.

Die zweite Seite der Medaille ist ebenso wichtig: Was ist zwischen dem letzten Mal, als noch alles funktionierte und jetzt mit dem Rechner bzw. dem Programm passiert ? Auch hier das eine oder andere Beispiel:

- Rechner zur Reparatur - tut nicht. Selbst das Grundgerät will nicht. FDX ist jedoch O.K. Nach langem Suchen und Messen mit dem Oszilloskop sowie dem Zerkratzen von Leiterbahnen, um herauszufinden, wo wo aus das Datenbit an der CPU auf Masse gezogen wird förderte zu Tage, daß das IC zum Druckerausgang durchgebrannt war (und gleich zwei Nachbarn mitriß). Ein Anruf beim Absender dieser Kiste bestätigte meine Vermutung, daß er das Druckerkabel verkehrtherum draufgesteckt hatte - und sich am Rande wunderte, warum der Drucker nicht wollte. Da jedoch der Rechner mittlerweile seinen Geist aufgegeben hatte, ist er dem Druckerproblem nicht nachgegangen. Hätte ich das gewußt, wäre der Fehler viel schneller gefunden gewesen.
- Läuft ein Programm nach einigen Änderungen nicht mehr, sieht man sich ja auch diese Änderungen an, macht sie ggf. rückgängig, und baut sie schrittweise neu ein.

Ich hoffe, daß meine o.g. Beispiele etwas aufgezeigt haben, daß es zur Analyse eines Fehlers häufig hilft, zum einen dem, der den Fehler finden soll alles zusammen mit einem Lebenslauf des Patienten in die Hand zu drücken.

Praktisch bedeutet dies:

1. Läuft der Memotech nicht, immer Grundgerät plus Floppystation zur Reparatur geben! Und aufschreiben, was man so mit dem Teil verbrochen hat. Insbesondere ob man mal mit dem Lötkolben in der schwarzen Kiste unterwegs war, das eine oder andere umgestöpselt hat, den Netzstecker verkehrtherum in die Steckdose gesteckt hat u.s.w.
2. Läuft ein Programm nicht, bitte eine Diskette zusammenstellen, auf der all das drauf ist, um damit zu booten und den Fehler nachvollziehen zu können. Es muß nicht alles automatisch ablaufen. Ein BRIEF. auf der Diskette mit den erforderlichen Informationen genügt. Und bitte nicht nach dem Motto 'NewWord habe ich nicht mit draufkopiert, das wirst Du sicherlich zur Hand haben'. Wirklich ALLES auf die Scheibe!

C l u b: Korrektur & Nachtrag / Turbo-Tip / S o f t w a r e: Mtx-Edit

Korrektur und Nachtrag

E45.KLX (Herbert zur Nedden, 2000)

Leider ist mein MTX-Edit in der KLICK-Version etwas hinterhältig, da er vor jedem Diskettenzugriff ein allgemeines Disk-Reset - also BDOS-Funktion 13 - absetzt. Das sollte er sich schleunigst abgewöhnen - zumal ein R/O-Fehler ihn nicht weiter schreckt! Daher ist folgendes zu Patchen - hier gleich in der Notation für ThePatch:

Patch der KLICK-Version des MTX-Edit
E45.KLX

```
0
006BE 00 00 00 ; BDOS-Call für Drive-Reset aus-NOPpen
```

Info 27, S. 29ff: (Herbert zur Nedden, 2000)

Im Interrupt-Acknowledge Zyklus ist IOREQ=Low, M1=Low, jedoch entgegen der Angaben im Info 27 RD nicht =Low, sondern RD=High!

Cache und EPSON-Laufwerke: (Hartmut Traber, 5270)

Um an der Leitung 34 das Disc-Change-Signal zu erhalten, damit der Cache dieses auswerten kann, sind folgende Löt-Jumper zu ändern:

Jumper:	J8	J9	J10	J15
Alt:	auf	zu	auf	zu
Neu:	zu	auf	zu	auf

Turbo-Tip

(Herbert zur Nedden, 2000)

Zeileneditor unter Turbo-Pascal aus Info 33, S. 5 (Peter Lang, 8502)

1. Vor Aufruf sollte der Cursor an die richtige Stelle positioniert werden z.B. mit GotoXY.
2. SHIFT-Esc ist bei laufendem Zeileneditor tödlich!
Anm.d.HzN.: Das kann ich nicht bestätigen! Bei mir tut alles weiterhin.

MTX-Edit

(Herbert zur Nedden, 2000)

Jan Bredereke und Holger Hansen hatten einige Anmerkungen zu MTX-Edit in der KLICK-Version, auf die ich hier kurz eingehen möchte:

Unter MTX-Edit müssen die Zeilenenden immer aus CR-LF, d.h. ^M^J bestehen. Beim Laden der Datei stellt MTX-Edit dies sicher. Wer nun (wie Holger) alle ^J löscht kassiert leicht einen Absturz.

Die KLICK-Version ignoriert ^K^X. Sichern des Textes geht nur via ^K^S, Verlassen via Esc. Holger wollte gerne, daß ^K^X ein ^K^S und Esc bewirkt. Tja, ich habe es aus zwei Gründen nicht drin: Kein Platz im Help-Menü und ich hasse KLICKer, die ich so mirnichts-dirnichts verlasse. Folglich ist das Absicht!! ^K^X auf ^K^S zu legen fand ich nur irritierend. Außerdem merke ich so immer, in welchem MTX-Edit ich bin.

Ja, Holger, irgend wann lernen sich wohl DiJey und MTX-Edit kennen: Aufruf von MTX-Edit aus DiJey. Nur was mache ich mit altem Inhalt von MX-Edit ?

Nach ^K^D gilt der Puffer unter MTX-Edit nicht mehr als gerettet. Das ist für mich logisch, da immerhin nun der Puffer einer anderen Datei zugeordnet ist, und unter diesem Namen ist er ja auch noch nicht gerettet worden.

C l u b: Fragen & Antworten

Fragen und Antworten

- A: (Herbert zur Nedden, 2000)
Info 30, Seite 33: SuperCalc verbiegt den WarmBoot auf sich selbst, weshalb SHIFT-ESC W <RET> kein warmen Boot, sondern SC liefert. Was die Idee der SC-Hersteller ist, sich so einzuhängen, ist mir unklar.
- F: (Peter Lang, 8502 und Hartmut Traber, 5270))
Warum blinken nach einem CONFIG x:?? alle Laufwerke, die im Cache als automatische Laufwerke installiert sind, einmal kurz auf?
- A: (Olaf Krumnow, 2050)
Dieses Phänomen verwirrte mich zunächst auch etwas. Es tritt in Verbindung von Cache und CONFIG auf. CONFIG ruft (vermutlich) die BIOS-Funktion 'Select Drive' auf, um die Adresse des DPB (Disc-Parameter-Blocks) zu erhalten, der unter RAM 4.x auch Informationen über den Config-Code des Laufwerks enthält. Diese Funktion macht üblicherweise nichts anderes, als sich das Laufwerk zu merken und die Adresse des DPB zu liefern. Wenn der Cache installiert ist, ist diese Funktion aber erweitert und prüft bei automatischen Laufwerken, ob die Diskette gewechselt wurde. Dazu wird das Laufwerk kurz angewählt, die Wechsellinformation gelesen und das Laufwerk wieder abgewählt. Das geht so schnell, daß die Leuchtdiode am Laufwerk normalerweise nicht mal aufblinkt. Läuft das Laufwerk aber schon (z.B., weil CONFIG mit ?? die Laufwerke gestartet hat), so blinkt die Leuchtdiode doch auf. Das ist also alles völlig harmlos und leicht erklärbar. Überhaupt neigen einige Programme unter dem Cache zu einer wahren Blinkorgie. So veranstaltet z.B. BACKitUP beim Lesen von der Quelldiskette ein grandioses Konzert der Leuchtdioden. Sieht ulkig aus!
- F: (Andreas Fischer, CH-4303)
Wer hat eine Modem-Software für 1200/1200 Baud - BTX-Anschluß
- F: (Hauke Ahrensfield, 3100)
Kann man auch bei NW V2.12 irgendwie (?) das Laufwerk, wo sich die zu bearbeitenden Dateien befinden, vorab einstellen?
- A: (Herbert zur Nedden, 2000)
Als Notlösung: Ruf NewWord von dem Laufwerk aus auf, auf dem die Dateien sind. D.h. NW auf B:, Dateien auf C:, dann auf C: einloggen und dann C>B:NW .
- F: (Hauke Ahrensfield, 3100)
Bei 534 Sätzen (je 240 Byte) wollte DBase V2.41 die Anzahl der Sätze nach append bei mir nicht weiter erhöhen. Nachdem ich per DU2 diese selbst nachträglich erhöht hatte, lief alles wieder normal, auch bei weiterem 'Anhängen'. Normal, oder ist da auch irgendwo der Wurm im Prg?
- F: (Hauke Ahrensfield, 3100)
Auf dem letzten Clubtreffen hatte jemand 'n Multisync GS (?) an seinem MTX hängen. Wer war das nochmal? Hat einer Erfahrung mit dem ATARI SM124 am MTX (Aufgabe der einzelnen Steuerleitungen)?
- A: (Herbert zur Nedden, 2000)
Multisync: manfred Flume, 5600 Wuppertal
- F: (Hauke Ahrensfield, 3100)
Gibt es eigentlich schon 64K-SRAM-Chips? Wenn ja, wo für wieviel? Wo gibt es 32K-SRAM am billigsten?
- F: (Holger Hansen, 3300)
Hat vielleicht schon jemand DU2 oder DU3 so gepatcht, daß das Mapping bei initialisierten Disketten funktioniert, ohne Bildschirmmüll?
(initialisiert = mit Zeiteinträgen versehen)
- F: (Holger Hansen, 3300)
Hat schon jemand ein Compress Programm geschrieben?
(Compress: ordnet die Dateien auf Diskette/Festplatte so, daß die Daten der Dateien in aufeinanderfolgenden Sektoren stehen)
- F: (Herbert zur Nedden, 2000)
Wie bringe ich die Druckertreiber NWP216.OVR und NWP217.OVR von CLUB.040 mit NewWord Version 2.02 ans laufen?

Leserbrief: Claudio Romanazzi, 3070

Nun zu einer Vorausversion meiner neuesten Schöpfung - dem CACG. Das ist keine Schweinerei, sondern heißt auf deutsch Computer Aided Crossword Generator oder noch deutscher Computer gestützter Kreuzworträtsel-Erschaffer. Ich habe mir überlegt, was man alles mit der Bibliothek meines Syntax-Checkers (-Prüfers) anfangen kann. Dabei kam ich auf diese Anwendung. Vorläufig läuft noch gar nichts. Was Ihr in diesem Info seht, ist meist (schweißtreibende) Handarbeit. Zu Weihnachten wird die Version 1.0 wohl fertig sein. Diese Version ist von der Bibliothek völlig unabhängig. Eine weitere Version 2.0 wird dann sogar ohne menschliche Beteiligung zur Generation (Schöpfung) eines Kreuzworträtsels fähig sein. Nur die Hinweise müssen natürlich per Hand eingegeben werden. Ein Beispiel (entschuldigt das Design) kommt wohl auf den folgenden Seiten (nicht wahr, Herbert??). ((JA, CLAUDIO!!))

Bei dieser Gelegenheit fordere ich noch einmal alle Besitzer von Wortmanager auf, die inzwischen doch wohl weiter expandierte Bibliothek einzuschicken (kein Rückporto beilegen). Weiterhin möchte ich noch um Anregungen und Kritik bitten. Daraus wird Version 1.8 entstehen, die letzte nämlich. Danach wird Wortmanager in ein Textsystem einfließen und als Einzelprogramm nicht mehr gepflegt werden. Dieses Textsystem wird die automatische Silbentrennung beherrschen (von den 55 000 Wörtern der Grundversion der Bibliothek kann eine Vorversion (die erscheint auf einer der nächsten PD) 54 800! Wörter richtig trennen). In der Bibliothek werden nicht korrekt trennbare Wörter extra markiert, so daß eine Warnung erscheinen kann. Soweit meine Pläne.

Ebenfalls auf einer der nächsten PDs wird ein Pascal-Include-File zur Mathematik mit komplexen Zahlen sein. Dieses File ist noch nicht ganz fertig, kommt aber noch dieses Jahr.

Nun noch ein letztes Thema. Da ich meine Apfelmännchenpläne noch immer verfolge, habe ich mir überlegt, wie die doch sehr aufwendige Berechnung (siehe Include-File komplexe Zahlen) zu beschleunigen ist. Lösung: Aufbau einer 8087-ECB-Bus-Karte. Der 8087 ist bekanntermaßen ein Mathematik-Prozessor. Der ist so schnell, daß, einmal mit Daten beschickt, das Ergebnis praktisch sofort abgeholt werden kann. Die grundsätzliche Schaltung steht bereits, ein Prototyp wird (handverdrahtet) im Oktober ('89) aufgebaut werden. Wer, also Frage ich euch, hat noch Interesse? Man kann durch Sammelbestellung einiges Geld sparen. Der Prozessor mit 5 Megahertz (das reicht) kostet als Einzelstück ca. 175.- DM, mit allen Bauteilen ca. 260.- DM. Ich schätze, die Karte ist im März '90 incl. der notwendigen Software (natürlich Maschinensprache) fertig.

Anm.d.OK & HzN: Wir haben Claudio vorgeschlagen den schnelleren und schöner programmierbaren 68881-Coprozessor in Erwägung zu ziehen. Er hat nur einen Nachteil: er kostet ca. DM 70.- mehr, was er aber mit Qualität wett macht.

Claudio

Ach ja, was mir noch am Herzen liegt: DiJey ist super, wenn...

Bei mir kommt es eben nicht so oft vor, daß ich aus irgendeinem Programm mal raus muß, schnell was kopieren will oder etc. Statt dessen wünsche ich mir ein wirkliches Universalprogramm, daß auch wirklich alles erschlägt. Mich stören die Pips, Mcopys, Nsweeps etc. Es sollte ein Programm für alles geben! Ich sprach auch schon mit Olaf darüber, doch der stönte und beklagte den mangelnden Puffer, der dann entsteht, wenn das Programm immer länger wird. Ich speziell brauche ein Kopierprogramm, welches mir, fahre ich ein System hoch, Files von A nach B transportiert, damit, wenn meine Frau einen Text schreiben will, sie kein MTX-Spezialisten-Studium betreiben muß. Menu macht das für mich, doch oh weh, DeJey nicht. Also Mcopy her, das machts. Schön wäre also, oh Olaf, ein Miniprogramm, welches eine Kommandozeile schluckt und dann DeJey im Klick zu Hilfe ruft. Das wäre besser als ein dickes Mcopy auf der SRAM. Okay ???

K o m i k: 'n Kreuzwort-Rätzel

Anm.d.HzN.: Hier Claudio's Kreuzworträtzel - hat er mit extra frischen (und daher fettem) Farbband ausgedruckt. Beim Einkleben habe ich mir doch fast schwarze Flossen geholt!

11	1	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000
----	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

K o m i k : 'n Kreuzwort-Rätzel

WAAGRECHT

1. Schutzschicht
3. verfolgen
8. hervorragender Computer
9. Schafsfleischbraten
12. Wurstsorte
13. Speichermedium
16. Stadt an der Ostsee
18. unklar
20. Singvogel
23. Fragewort
24. nachgiebig
26. Seelenhauch
27. Gesichtsschutz
29. Entfernungseinheit (Abk.)
30. Verwegenheit
31. Anlage zum Schmelzen von Eisen
33. Rauschgift
34. Hochschulleiter
35. Gesellschaftsform (Abk.)
37. sei schnell!
39. Augenblick
40. vorher
41. Reptil
43. Gewinner
44. grosses Gebaeude
45. asiatischer Staat
46. germanischer Speer
47. Grundnahrungsmittel
48. Fachausstellung
50. erster Mensch
53. Gefaessboden
56. Fernsprechapparat
58. Tier mit langem Hals
59. engl. nein
60. Spitzname des Hasen
62. Meteorologe
67. Zeitmass
68. Vogelprodukt
70. Vorfahr
71. Farbe
72. Brot Gottes
74. Kuechenchef
76. beleuchtet
79. Geschlechtskrankheit
80. Tischauflage
81. Bodenvertiefung
82. Stadt im Ruhrgebiet
84. Mahlzeit fuer Tiere
86. bekannte Floppystation
87. betteln
89. Raketenmetall
91. Einbeulung
93. toter Koerper

94. Erhebungen
96. grosser Raum
98. Schock
100. Flaechenmass
101. vorbildlicher Junge
103. Haustier
104. Autokennzeichen Seesen
105. griechischer Buchstabe
107. Rumpfstueck
108. chinesische Philosophie
111. Schrift am Kreuz
112. Frauennamen
113. Mitglied der NSDAP
114. Geschehensfolge
116. Maennernamen
118. Mehlspeise
119. Haargleichrichter
120. Sitz der Clubleitung
121. Priester (Mz)
122. lat. und

SENKRECHT

2. Hauptstadt der Ukraine
3. Zustimmungswort
4. betagt
5. fertiggekocht
6. Laufvogel
7. zu keiner Zeit
8. Koenner
9. Salamander
10. Brei
11. Verhaeltniswort
12. bereits
14. Singvogel
15. Windstoss
16. antike Waffe, Sportgeraet
17. Vielleser
19. Gutschein
21. internationales Notzeichen
22. Gewichtsklasse
23. Geisteskrankheit
25. Reptil
28. Zeittafel
32. waagrecht
34. Karateverein
35. Faultier
36. durchsichtiger Stoff
38. Musikstueck
42. Grottentier
49. teuflisch
51. Rindervorfahr

52. mehrmals
54. Stachel
55. Stadt an der Donau
57. erste Saengerin
58. meisterhaft
61. duenn
63. Nadeloer
64. Gebraeuche
65. Jahreszeit
66. schmales Brett
67. geistig
69. eisern
72. Vererbungsforscher
73. Spielkarte
75. Nebenfluss der Donau
76. Flugzeug
77. Clubleiter (Nachname)
78. Name eines Weltraummonsters
83. Metall
85. aromatisches Getraenk
87. Witz
88. Wiesenzaun
90. Clubleiter (Vorname)
92. fluessiges Gestein
93. Gartenschaedling
94. Treibstoff
95. Einkehrer
97. Nordseeinsel
99. steiler Felsabhang
102. Geraeusches einer Blase
106. Bildschirmart
108. Zeiteinheit
109. Verhaeltniswort
110. spanische Welle
112. Scherz
115. amerik. Nachrichtenagentur (Abk)
117. Skatausdruck

Leserbrief: Jan Bredereke, 2000Jan
Bredereke

Hamburg, den 26. August 1989

Bauernvogtkoppel 65 a
2000 Hamburg 65
Tel.: (040) 601 40 07
Postgiroamt Hmb BLZ 200 100 20
Postgirokto.-Nr.: 5158 94-204Jan Bredereke, Bauernvogtkoppel 65a, 2 HH 65MTX-User-Club
c/o Herbert zur Nedden
Sonnenau 2

2000 Hamburg 76

Hallo Herbert!

Das Info 33 und die KCLICK.007 (007 = J.B. = <siehe Kopfzeile, Name und Telefonnummer>) waren mal wieder ganz toll und informativ. All die schönen neuen Sachen haben mich erst mal wieder durch Lesen, Installation usw. von meinen eigenen *.MAC abgehalten, aber jetzt ist es geschafft. Dazu noch was Kurzes für's Info:

Disketten-Formate

(Jan Bredereke, 2000)

In Info 32-16 hatte ich meine Probleme beim Formatieren mit meinem selbstdefinierten Diskettenformat FB (wie 09, nur ohne Systemspuren) beschrieben. Nun kam mir gerade die KCLICK.007-Diskette ins Haus mit FORMSTAR von Michael Keßler. Beim Ausprobieren entdeckte ich zu meiner großen Freude, daß FORMSTAR jetzt prompt auch mein Format FB unterstützt. Vielen Dank, Michael!

Daher für alle, die auf ihren 09er-Scheiben die Systemspuren dank einer SRAM-Disk nie brauchen, noch einmal die genaue Formatbeschreibung, um nicht verschiedene Versionen zu erzeugen:

Alles ist wie bei 09, bis auf:

- + 0 Systemspuren statt 2, dadurch keine Platzverschwendung für Systemspuren
- + 128 Directory-Einträge statt 64, da mir das immer zu wenig war, denn die Datumseinträge fressen immer ein Viertel aller Einträge.

Die geänderten Werte für FORMATE4 im einzelnen:

- * Name: "MaxJB 5" 40T 0Sys 128Dir"
 statt "MaxBP 5" 40T 2Sys"
- * Bloecke: 200 statt 195
- * Dir-Einträge: 128 statt 64
- * ALO: #C0 statt #80
- * CKS: 32 statt 16
- * OFF: 0 statt 2

Für das Formatieren ist die Anzahl der Directory-Einträge aber egal, nur die Formatkennung ist anders und muß anders eingetragen werden, wie schon in Info 33-23 beschrieben ist: Bei FB wird die Kennung mit INITDIR+ von KCLICK.007 zwischen die Datumseinträge gequetscht.

Leserbrief: Jan Brederke, 2000

Hier gibt es allerdings auch noch ein Problem: Die automatische Formaterkennung von CFG5.COM/KLX erkennt die Kennung dort noch nicht, obwohl sie da ist.

Und wenn ich schon einmal dabei bin: Ich habe vor einer Weile das Format des Kaypro 10 herausgefummelt, weil ich mit einem Daten austauschen wollte. Hier ist das Format. Eventuell sind die Sachen, die das Formatieren betreffen, nicht ganz richtig, aber Lesen und Schreiben funktioniert.

```

Format-Nummer:   vorläufig FD
Name:            Kaypro 10
SPT:            40
BLOCKSIZE in K: 2
Bloecke:        197
EXM:            1
Dir-Einträge:   64
AL0:            #C0   <== Hier muß FORMATE4 protestieren, denn es wäre
                   Platz für 128 Dir-Einträge, von denen aber
                   tatsächlich nur 64 genutzt werden.

AL1:            #00
CKS:            16
OFF:            1
RECLN:         512
FIRST:         0
TRACKCNT:      40
Seiten:        2
Density:       2
5 oder 8 Zoll: 5
VR/RR:        0
Invertiert:    0
TrackTyp:     10
SKEW:         1
phys. Sekt./Spur: 10

```

Noch ein paar Anmerkungen zur Ankündigung von RAM5

Ich wünsche mir sehr 24 oder 32 Funktionstasten für Klick-Programme, mit den 16 komme ich schon jetzt nicht aus! Wie man ^F2' und ^F8' mit Dioden entkoppelt, habe ich ja schon in einem früheren Info beschrieben.

Die Möglichkeit, beim Start von RAM die Klick-Programme nicht zu löschen, ist eine feine Sache, besonders für E45 und NOTE, wenn sie noch Texte geladen haben. Allerdings sollte man bedenken, daß sich manche Programme z.B. in den Uhren-Interrupt oder andere Vektoren einhängen. Diese Programme werden durch das neue Booten dann trotzdem in seltsame Zustände gebracht, weil sie nicht korrekt deinstalliert werden.

Viele Grüße

Jan Brederke

Leserbrief: Hans Gras, NL 1506 / R A M 5.x: ZCPR3

Hallo, Herbert

Diesmal wieder einige Sachen aus Zaandam.

- ZCPR3 läuft zeit einigen Tagen fabelhaft auf meiner Kiste. Mit RAM 4.5 in einer 58K Umgebung, das heißt, daß ich keine kleinere TPA habe. Ein Teil des Puffers ist unterhalb TOAM abgespeichert. Das macht mir garnichts aus, ich habe mit (nur) 4 Laufwerken und eine RAM Floppy noch einige Bytes übrig.
- Nur die .RCP ist nicht implementiert, bringt auch nichts, wir haben KLINK und RAM-Disks u.s.w. und das reicht.
- Wegen SHells (WordStar 4.0 is auch einer!) gibt es mit dem WarmBoot aus KLINK einige Probleme: die SHells werden wieder aktiviert. Etwas für RAM 5.0 um zu lösen?
- Meine IO Package ist kleiner. Nur 5 Sektoren, aber wir können neue Packages nachladen. Also, auch kein Problem.
- Wenn es für jemand intressant ist, um mit ZCPR3 zu üben, bis RAM 5 fertig ist: Bei mir melden.
- Alle Library Utilities (ZCPR2) habe ich auch geändert. War nicht unbedingt notwendig aber macht Spaß, von ZCPR2 in ZCPR3 umzubauen. Die Z3SYSLIB ist Spitze. Mehrere Programmen in ZCPR3 sind (viel) kürzer geworden.

Hans Gras

ZCPR3 - Was bietet es ?

(Holger Hansen, 3300)

Die folgende Übersicht ist eine kurze Zusammenfassung aus dem ZCPR3 User Manual und dem ZCPR3 Installation Manual. In Anbetracht eines Info Artikels natürlich nur ein Anriß des Themas. Ein vollständigerer Überblick ist im User Manual zu finden, wenn auch in englischer Sprache. Die Tabellen am Ende sind so etwas wie eine Schnellübersicht (neudeutsch: Quick Reference Guide).

Für alle diejenigen deren Verständnis schon bei ZCP.. aufhört, nur ganz kurz folgendes. ZCPR3 steht für Z80 Command Processor Replacement Version 3. Ein Command Processor ist derjenige Teil vom CP/M, der die Benutzereingaben auswertet und dementsprechend handelt. Z80 heißt, daß dieser ZCPR3 in Z80 Code geschrieben ist, was effektiver als CP/Ms 8080 Code ist. Replacement Version 3 heißt, daß der ZCPR3 den CCP bzw die Version 2 des ZCPR ersetzt. (ZCPR2 wird z.B. bei RAM 4.x mitgeliefert)

Für alle diejenigen, die mehr wissen möchten, empfehle ich die ersten vier Literaturstellen.

1. Prinzip und Übersicht

ZCPR3 besteht aus einem Kern dem eigentlichen Command Processor und einer Reihe von Puffern, die außerhalb des ZCPR3 liegen. Der Inhalt der meisten Puffer läßt sich zur Laufzeit mit dem Programm LDR.COM ändern.

Folgende Programmteile gibt es:

- "Environment" Paket
- "Flow Command" Paket
- "Input/Output" Paket
- "Resident Command" Paket
- "Named Directory" Paket
- "Multiple Commandline" Puffer
- "Shellstack" Puffer
- "Message" Puffer

Paket ist hierbei zum einen, eine ladbare Routine und zum anderen ein, bei der Erstellung des ZCPR3 festinstallierter, Speicherbereich. Die Routinen stehen in Dateien mit, für jedem Paket typischer Datei-Endung und können zur Laufzeit in den jeweiligen Puffer geladen werden.

R A M 5.x: ZCPR32. Im Einzelnen (in Klammern die Dateieindung)

2.1. Das Enviroment Paket (ENV)

(zu Deutsch Umgebungspaket) enthält Angaben über die Zahl, Art und Größe der ZCPR3 Puffer, sowie eine Reihe von Terminaleigenschaften, wie Spaltenzahl, Zeilenzahl oder Cursorpositionierung. Über diesen Puffer kann ein Programm alle wichtigen Informationen über das System erhalten. Mit dem Programm LDR.COM kann man den Puffer jederzeit verändern und somit z.B. die Terminaleigenschaften ändern, ohne die Utilities ändern zu müssen. In den Utilities steht nämlich nur ein Zeiger auf diesen Bereich.

2.2. Das Flow Command Paket (FCP)

(in etwa "Ablauf Kontrolle") ist eine Routine, die die erweiterte Ablaufkontrolle enthält. Die "Flow Commands" sind IF, ELSE, FI und XIF. Sie entsprechen denen von vielen Hochsprachen bekannten IF-Strukturen, wobei FI einem ENDIF (Beende IF) entspricht. XIF steht für "Beende alle IFs. Diese Befehle steuern den Kommandoablauf durch das Setzen oder Nichtsetzen des sogenannten FLOW STATUS. Dieser kann den Wert True (=wahr) oder False (=falsch) annehmen. Wenn der Status True ist werden alle Kommandos bis zum nächsten Flow Command ausgeführt, falls er False ist, werden alle folgenden Befehle, bis auf Flow Commands, übersprungen. Diese Eigenschaft ist vorallem für ZEX und SUB Dateien vorteilhaft. Die Bedingungen, die den Flow Status beeinflussen, stehen in Tabelle 5.

```
Bspl.: A>IF EM TEST.FAT      "wenn die Datei TEST.FAT leer ist"
        A> DDT TEST.FAT      "führe dies aus
        A>ELSE              "falls nicht leer"
        A> EDIT TEST.FAT    "führe dies aus"
        A>FI                 "beende Flow Command"
        A>DIR TEST.FAT      "führe unter allen Umständen aus"
```

Wenn in diesem Bspl. TEST.FAT leer gewesen wäre, dann würden die Befehle DDT TEST.FAT und DIR TEST.FAT ausgeführt, andernfalls EDIT TEST.FAT und DIR TEST.FAT.

2.3. Das Input/Output Paket (IOP)

Gibt es ja schon seit ZCPR2. Deshalb hier nur dieses: das IO-Paket ist eine Sammlung von Ein- und Ausgaberroutinen, die dazu benutzt werden, dem BIOS grundlegende Routinen zur Konsolen, Drucker, Reader und Puncher Ein- und/oder Ausgabe zur Verfügung zu stellen. Dieses IOP kann mehr Treiberroutinen für die Kanäle verwalten, als CP/Ms IO-Byte.

2.4. Das Resident Command Paket (RCP)

RCP ist ein Speicherbereich indem kurze Routinen abgelegt sind, die die in ZCPR3 fest eingebauten Kommandos (Tabelle 1) ergänzen oder entsprechende COM Dateien ersetzen. Der Vorteil der RCP Kommandos ist der schnellere Zugriff (keine Diskettenladezeiten) und die Einsparung von Diskettenplatz. Für RAM 5 Benutzer sind diese RCP aber besser durch KLIKK Overlays realisiert.

2.5. Named Directory Paket (NDR)

Auch aus ZCPR2 bekannt. Named Directory (=benanntes Inhaltsverzeichnis) ist ein Synonym für ein logisches Directory unter ZCPR3. Logische Directories sind Kombinationen aus Laufwerk und Userbereich. Auf jeder Diskette gibt es also maximal 32 logische Direktories. ZCPR3 ermöglicht es nun jedem dieser Directories einen Namen zu geben. Bspl. Laufwerk A User 15 ist A15: (DU-Form) oder ROOT: (Named-DIR-Form).

2.6. Multiple Command Line Puffer

MCL (ungefähr "Mehrere Befehle pro Zeile") ist ein Speicherbereich in dem die Eingabezeile abgelegt wird. Wie schon bei ZCPR2 können mehrere durch Semikolon getrennte Befehle in einer Zeile stehen. ZCPR3 stellt hierzu noch den internen Befehl NOTE zur Verfügung, der Kommentare innerhalb solcher Befehlszeilen erlaubt.

R A M 5.x: ZCPR3

2.7. Shell Stack

Ist ein Speicherbereich indem Namen und Status einer Shell abgelegt sind. Wenn innerhalb einer Shell eine weitere aufgerufen wird, wird die alte Shell auf diesem Stack abgelegt und von dort zurückgeholt, wenn die neue Shell wieder beendet wird. Was ist eine Shell? Die bekannteste Shell ist MENU.COM (zumindestens bei RAM4 Benutzern). Shells sind eine Schnittstelle zwischen ZCPR3 und dem Benutzer. Hinter einem Tastendruck können eine ganze Folge von ZCPR3-Kommandos stehen.

2.8. Message Puffer

Ist ein 80 Bytes großer Speicherbereich, in dem Programme und ZCPR3 Informationen ablegen oder auslesen können. Bspl.: Ist eine Shell aktiv oder ist ZEX aktiv oder welcher Error Status anliegt. Außerdem gibt es hier 9 Bytes als 9 1-Byte Register, in denen Parameter übergeben werden können.

2.9. Suchpfad

Ist eine Reihe von logischen Directories in denen nach COM Dateien gesucht wird. Bspl.: Pfad A0: A12: H0: I0: bedeutet, daß der Reihe nach auf Laufwerk A User 0, Laufwerk A User 12, Laufwerk H User 0 und zum Schluß auf Laufwerk I User 0 gesucht wird.

3.0 ZCPR3 und RAM 5

ZCPR3 kostet gegenüber ZCPR2 ca. 1.5 kBytes mehr Speicher. Darum muß im Commonbereich von RAM 5 erstmal einiges an Platz geschaffen werden. Was von ZCPR3 letztendlich wirklich eingebaut wird, entscheiden die Konstrukteure in und um Hamburg. Ich hoffe Herbert hat in Tabelle 2 hinter allen später verwirklichten Features ein Kreuzchen gemacht. Zu RAM 5 kann ich natürlich ebenfalls keine Angaben machen, nur daß ZCPR3 nur in Verbindung mit RAM 5 ausgeliefert wird.

4.0 Literatur:

1. CP/M Anwenderhandbuch	Tom Hogan	Osborne/McGraw-Hill	D
2. Das Handbuch des CP/M 2.2 Betriebssystem	Digital Research	Markt & Technik	E
3. CP/M Handbuch mit MP/M	Rodney Zaks	Sybex	D
4. Programmentwicklung unter CP/M 2.2 für CPC 464/664	Helmut Tischer	Markt & Technik	D
5. ZCPR2 Konzept Manual	Richard Conn		E
6. ZCPR2 User Manual	Richard Conn		E
7. ZCPR3 User Manual	Richard Conn		E
8. ZCPR3 Installation Manual	Richard Conn		E
9. ZCPR3 Help File System	Richard Conn		E
10. RAM 4 Handbuch S. 38-41	Bernd Preusing		D

Hinzu kommt an Literatur:

11. ZCPR3-Einführung ... Übersetzt von	Richard Conn Hartmut Traber		D
12. RAM 5 Handbuch	O. Krumnow & H. zur Nedden		D

(D Literatur in Deutsch, E Literatur in Englisch)

R A M 5.x: ZCPR3Tabelle 1:

Interner Befehl	Bedeutung	CCP	ZCPR2	ZCPR3
D:	logge in Laufwerk D ein	x	x	x
U:	logge in User U (0..15)		x	x
DU:	logge in Disk D User U ein		x	x
USER n	logge in User n (0..15)	x		
DIR <<DU:>afn> <AS>	Zeige Inhaltsverzeichnis nach Muster *.* (ohne Angabe) oder nach Muster DU:afn. S = zeige SYS Dateien A = SYS und DIR Dateien	x ¹	x	x
TYPE <DU:>ufn <P>	Zeige Datei ufn am Bildschirm P = schaltet seitenweise Ausgabe ein oder aus.	x ¹	x	x
LIST <DU:>ufn <P>	Ausgabe Datei ufn auf Drucker P = siehe TYPE		x	x
ERA <DU:>afn <I>	lösche Datei afn I = mit Sicherheitsabfrage	x ¹	x ²	x
REN <DU:>ufn2=ufn	Benenne Datei ufn in ufn2 um existiert ufn2 wird gefragt, ob ufn2 gelöscht werden soll	x ¹	x	x
SAVE nn <DU:>ufn <S>	Speichere ab 0100h nn * 256 Bytes in Datei ufn ab. nn als dezimal Zahl S = statt nn * 256, nn * 128 Bytes	x ¹	x	x
SAVE nnH <DU:>ufn <S>	dito mit nn als Hex-Zahl		x	x
GO parameter	Springe nach Adresse 0100H und führe Programm dort aus übergebe dabei parameter		x	x
JUMP adr	Springe nach Adresse adr 0..FFFF		x	x
GET adr <DU:>ufn	lade Datei ufn nach adr (Art Umkehrung des SAVE Befehls)		x	x
NOTE	"Tue nichts" Befehl (kennzeichnet Kommentar)			x

Erläuterung:

DU: kann immer optional zum Dateinamen stehen U = 0..31
einloggen U = 0..15

ufn heißt Dateiname ohne Wildcards (*,?)

afn heißt Dateiname incl. Wildcards

adr ist eine Hex-Adresse 0000..FFFF

nn ist eine Zahl im Bereich 0..255

nnH ist eine Zahl im Bereich 0..FF

<> heißt nur optional, muß also nicht angegeben werden
(<> übrigens nicht mit eingeben)

)¹ für die CCP Version gilt nur die Angabe des Laufwerks ohne User
und keine der Optionen (S,P,I)

)² Option bei ZCPR2 ist V, es wird nur für alle Dateien auf einmal abgefragt

drucken,

Drucker an das LST Gerät schicken, was immer sich dahinter verbirgt
Bildschirm schicke an CON Gerät

R A M 5.x: ZCPR3Tabelle 2: Die ZCPR3 Pakete

Name	Abkürzung	Filetyp	Eingebaut ?
Named Directories	NDR	*.NDR	wer will das ?
Input Output Package	IOP	*.IOP	ja
Resident Command Package	RCP	*.RCP	ja
Flow Command Package	FCP	*.FCP	evtl.
Environment Descriptor	-	*.ENV	ja!
Shellstack	-	-	ja
Messagepuffer	-	-	ja
Multiple Commandline	-	-	ja

Tabelle 3: Kommandohierarchie

Was passiert, wenn ein Befehl eingegeben wird ?

Kommandozeile: A>TEST PARAM1 PARAM2

- 1) die Zeile wird untersucht und Teile in diversen Puffern verteilt.
TEST in FCB1 Puffer ab 5Ch, param1 in FCB2 ab 6Ch
param1 und param2 in Puffer ab 80h
- 2) falls FLOW COMMANDS aktiv sind, wird TEST an FLOW Paket übergeben und je nach FLOW STATUS ausgeführt oder übersprungen und dann (9)
- 3) es wird geprüft, ob TEST ein gültiger RCP Befehl ist. Wenn ja -> ausführen, und dann (9), wenn nein,
- 4) ist es ein ZCPR3 interner Befehl? Ja -> ausführen und (9), nein, dann
- 5) wird entlang des Pfades nach einer Datei TEST.COM gesucht. Wird sie gefunden, dann wird sie in den Speicher geladen, gestartet und -> (9), wenn nicht dann
- 6) wird der Extended Command Processor (Datei CMDRUN.COM) gesucht, falls aktiviert, gestartet und ihm wird die Kommandozeile übergeben. Wenn nicht, dann
- 7) wird geprüft, ob ein Error Handler aktiv ist und ihm gegebenenfalls die Kommandozeile übergeben,
- 8) falls gar nichts zutrifft, wird die Meldung "TEST not found" ausgegeben
- 9) nächste Zeile -> (1)

Tabelle 4: Flow Bedingungen (FB)

IF EX(ist) ufn	prüft, ob Datei ufn existiert, ja -> TRUE
IF ER(ror)	prüft, ob ERROR FLAG gesetzt ist, ja -> TRUE
IF EM(pty) ufn	prüft, ob File ufn leer ist, ja -> TRUE
IF IN(put)	falls Taste T,Y,Space oder Cr gedrückt -> TRUE alles andere -> FALSE
IF NU(11) arg	stellt fest, ob arg definiert ist, ja -> FALSE
IF NU	gibt immer TRUE zurück
IF n val	prüft, ob Inhalt von Register n (0..9) gleich dem Wert val ist, ja -> TRUE
IF WH(eel)	Ist das Wheelbyte gesetzt? ja -> TRUE
IF TC(ap)	stellt fest, ob ein Terminal im Environment definiert ist
IF BFB	"B" Operator gilt für jede FB als NOT Verknüpfung. FBs sind: EX, ER, EM, IN, NU, WH, TC IF BEM TEST.DAT ist wahr, wenn TEST.DAT nicht leer ist.

Erläuterungen: (Die Buchstaben in Klammern nicht eingeben)

ufn Dateiname ohne Wildcards,

arg kann z.B. in SUB Dateien ein \$2 sein

IF NU \$2

ECHO Sie sollten den zweiten Dateinamen mit angeben

ELSE

EDIT \$2

FI

S u p e r C a l c : R a n g l i s t e m i t S C e r s t e l l e n

SuperCalc: Ene mene mu, raus bist du

(Kurt-Bernd Rohloff,8000)

Zusammenfassung: In diesem Beitrag zeige ich auf, wie man von einer Zahlenkolonne eine Rangliste nach absteigender Zahlengröße erstellen kann. Das Verfahren ist iterativ und ähnelt einem Abzählreim. Darauf aufbauend wird ein einfaches Sortierverfahren angegeben. Das allgemeinere Sortierproblem, einen Rechteckbereich nach den Werten einer bestimmten Spalte zu sortieren, wird in einem Folgebeitrag erörtert.

Häufig hat man das Problem, daß man eine Zahlenkolonne nach ihrer Größe geordnet dargestellt haben möchte. Dies wäre ja kein Problem, wenn SC einen Sortierbefehl hätte. Hat es aber leider nicht! Trotzdem ist man nicht ganz aufgeschmissen. Es wäre ja schon von Vorteil, wenn SC eine **Rangliste** dieser Zahlen erstellen könnte. Will sagen, die größte Zahl bekommt den Rang 1, die zweitgrößte den Rang 2 usw. Daraus kann man schon viel besser, sagen wir, die ersten drei oder die letzten drei ansehen als aus den ursprünglichen Zahlen. (Die Reihenfolge "von groß nach klein" ist von mir so gewählt, weil sie für meine Zwecke praktisch ist. Man könnte auch anders herum vorgehen und der kleinsten Zahl den Rang 1, der zweitkleinsten den Rang 2 usw. zuordnen.) Das Erstellen so einer Rangliste mit SC ist zwar auch nicht gerade ein "Pipifax", aber es geht. Wie man daraus eine "richtige" Sortierung machen kann, werde ich im nächsten Beitrag beschreiben.

Ich will das (recht trickreiche) Verfahren an folgendem Beispiel erklären:

	A	B	C	D	E	F
1	Rangliste			RANGDEMO	col	man
2						
3	Schalter	0				
4	Zähler	0				
5						
6		Daten		Hilf		Rang
7		=====				
8		4.54 !		4.54		0
9		5.62 !		5.62		0
10		5.1 !		5.1		0
11		4.84 !		4.84		0
12		5 !		5		0
13		4.38 !		4.38		0
14		4.5 !		4.5		0
15		7.8 !		7.8		0
16		4.44 !		4.44		0
17		4.02 !		4.02		0
18		=====				
19		Maximum:		7.8		0

Das Spreadsheet muß im Modus "manuelle Berechnung" und "spaltenweise Berechnung" sein, einzustellen mit /GM und /GC. Günstig, jedoch nicht unbedingt notwendig, ist außerdem der "next-off" Modus (/GN). In der Spalte B befinden sich unsere Daten, was immer sie bedeuten mögen. Für das Verfahren benötigen wir weiterhin diverse Hilfszellen. Eine mit "Schalter" bezeichnete Zelle dient dazu, die Rangberechnung ein- (B3=1) oder auszuschalten (B3=0). Zu Anfang muß B3=0 sein, damit die Hilfsspalten D und E richtig initialisiert werden. Außerdem benötigen wir in B4 noch einen Zähler. Er wird uns später die Ränge liefern. In B4 ist eingetragen: IF(B3=0,0,B4+1)

SuperCalc: Rangliste mit SC erstellen

D.h., solange der Schalter auf 0 steht, wird der Zähler auf 0 initialisiert. Erst wenn der Schalter B3 auf 1 (oder einen anderen Wert > 0) gesetzt wird, wird der Zähler bei jeder Neuberechnung des Spreadsheets inkrementiert.

Nun kommen wir zum wichtigsten Teil, den Spalten D und E. In E soll nachher die Rangliste erscheinen, D ist nur eine Hilfsspalte. Ich will vorab versuchen, das Verfahren in groben Zügen zu erläutern, damit du die folgenden Formeln besser verstehst. Das Verfahren berechnet in einem Rechenschritt lediglich einen Rang, man muß also so viele Berechnungen durchführen, wie man Daten hat. In einem Rechenschritt passiert folgendes: In der Zelle D19 wird jeweils der größte Wert der D-Spalte ermittelt. In der Rang-Spalte wird in der Zeile, in der dieser größte Wert in der D-Spalte vorkommt, der jeweilige Zählerstand eingetragen. Daraufhin wird im nächsten Schritt der größte Wert in der D-Spalte gestrichen und das Maximum von den verbleibenden ermittelt. Das Verfahren ähnelt also einem Abzählreim, wobei immer der größte Wert der Hilfsspalte ausscheidet.

In D8 steht die folgende Formel: `IF(E8>0,0,B8)`

Am Anfang ist $E8=0$, so daß in D8 der Wert von B8 eingetragen wird. Dies ist der oben dargestellte Anfangszustand. In den weiteren Zellen der D-Spalte stehen entsprechende Formeln, die wir aus D8 mit Hilfe des /R Befehls erzeugen können (`/RD8,D9:D17`). In D19 tragen wir ein: `MAX(D8:D17)`

Die Formel in E8 sieht etwas komplizierter aus:

`IF(B3>0, IF(AND(D8=D19,D19>0),B4,E8), 0)`

In E9 bis E17 tragen wir wieder mit dem /R Befehl entsprechende Formeln ein, wobei wir aber mit der Option A(sk for adjust) arbeiten müssen, denn B3, D19 und B4 dürfen hier nicht angepaßt werden. In E19 wird wieder das Maximum gebildet.

Jetzt wollen wir uns diese Formeln mal in Ruhe klarmachen. Zu Anfang setzen wir, wie gesagt, den Schalter B3 auf 0 und lassen SC zweimal rechnen (zweimal ! eingeben). Dann sollte das Spreadsheet so wie oben aussehen. Da $B3=0$ ist, liefert uns die Formel in E8 jetzt 0 (das ist die letzte Null in der Formel). Bei der zweiten Berechnung (spätestens) "sehen" die Formeln in der D-Spalte, daß rechts von ihnen (also E8 ...) eine Null steht, und liefern daher den Wert links von ihnen (B8 ...).

Nun kommt der große Augenblick, in dem wir den Schalter auf 1 stellen. Bei der nächsten Berechnung (ausgelöst durch !) erhöht sich zunächst mal der Zähler auf 1. In der D-Spalte passiert noch gar nichts. Aber in der E-Spalte erscheint in Zeile 15 die 1, bloß warum? Schauen wir uns dazu nochmal obige Formel für E8 an: jetzt ist ja B3 (der Schalter) größer als Null, so daß nun der IF-Ausdruck in der Mitte zum Tragen kommt. Der sieht für Zelle E15 so aus:

`IF(AND(D15=D19,D19>0), B4, E15)`

Vom Standpunkt der Zelle E15 aus gesehen heißt das: ist die linke Zelle von "mir" gleich dem Maximum in D19 und ist dieses Maximum auch größer als Null, dann trage bei "mir" den Zähler B4 ein. Daher die 1. In den anderen E-Zellen ist die Bedingung nicht erfüllt, so daß sie einfach ihren alten Wert wieder bei sich eintragen. Bei der nächsten Berechnung wird nun die Zelle D15 "ausgezählt", also auf Null gesetzt, weil jetzt ja in E15 eine Zahl > 0 (nämlich die 1) drinsteht. Daraufhin erscheint in D19 der nächst kleinere Wert 5.62 aus Zeile 9. Für E9 ist jetzt die Bedingung erfüllt, daher trägt sie bei sich den Zähler, der jetzt 2 ist, ein, alle anderen E-Zellen bleiben unverändert. Ich glaube, du ahnst jetzt, wie der Hase läuft. Wenn du das Spreadsheet 10 mal durchgerechnet hast, sollte es wie das folgende aussehen. Weitere Berechnungen stören nicht. Sie erhöhen lediglich den Zähler weiter.

Es gibt noch ein kleines Problem. Es könnte ja sein, daß in den Daten auch negative Werte vorkommen. Dann funktioniert das Verfahren so nicht. Man muß dann statt der Nullen, die ich in den obigen Formeln unterstrichen habe, eine negative Zahl einsetzen, die garantiert kleiner als alle möglichen Datenwerte ist.

SuperCalc: Rangliste mit SC erstellen

	A	B	C	D	E	F
1	Rangliste			RANGDEMO	col	man
2						
3	Schalter	1				
4	Zähler	10				
5						
6		Daten		Hilf		Rang
7		=====		=====		=====
8		4.54 !		0		6
9		5.62 !		0		2
10		5.1 !		0		3
11		4.84 !		0		5
12		5 !		0		4
13		4.38 !		0		9
14		4.5 !		0		7
15		7.8 !		0		1
16		4.44 !		0		8
17		4.02 !		4.02		10
18		=====		=====		=====
19		Maximum:		4.02		10

Hier noch ein kleiner Trick. Bei 10 Werten kann man die Berechnungen ja noch von Hand ausführen, aber bei 20 oder 30 ist das schon recht langweilig. Warum nicht ein Programm schreiben, das die notwendigen Berechnungen ausführt? Das geht einfacher, als du denkst, es ist nämlich schon im Spreadsheet drin!!! Es steht nämlich in Spalte C von Zeile 8 bis 17. Spalte C hat bei mir die Breite 3 (einstellen mit /FCC,3), obwohl das nicht von Belang ist. Wir brauchen nur den Bereich C8:C17 in eine Datei abzuspeichern, aber bitte **ohne** Rand:

```
/GB                               Rand aus
/ODC8:C17,DRUN.XQT               abspeichern in RUN.XQT
/GB                               Rand wieder ein
```

Vorgehen: Spreadsheet initialisieren (B3=0, !!), dann B3=1 setzen und Befehl /XRUN eingeben. Zurücklehnen und zuschauen.

Noch 'n Trick: Die gestrichelte Linie gibt man in B7 und B18 als '=' ein. Dann geht sie aber bis an den rechten Bildschirmrand. Um sie mit Spalte E aufhören zu lassen, gibt man in F7 und F18 ein Leerzeichen als Text ein (" ").

Wer mir bis hierher tapfer gefolgt ist, wird es nicht schwer haben, auch die (absteigende) Sortierung der Daten zu begreifen. Dazu beachten wir, das in der Maximum Zelle D19 die Daten ja nacheinander in absteigender Größe erscheinen. Wir müssen sie dort nur noch zum rechten Zeitpunkt abfangen. Das ist jetzt nur noch ein kleiner Schritt. Zunächst numerieren wir die Daten in Spalte A fortlaufend durch. In A8 tragen wir ein: A7+1 und in den weiteren Zellen entsprechendes (/RA8,A9:A17). Wenn wir jetzt in Spalte F Formeln eintragen, die immer dann, wenn der Zähler gleich der laufenden Nummer "ihrer" Zeile ist, den Wert von D19 bei sich eintragen, haben wir es schon geschafft. In F8 tragen wir also ein: IF(B4=A8,D19,F8) und in den restlichen Zellen entsprechendes, wobei B4 und D19 **nicht** angepaßt werden. Diese mit "Sort" überschriebene Spalte kann so jedoch nicht initialisiert werden. Wenn dies gewünscht wird, kann man die eben genannte Formel auch noch vom Schalter B3 abhängig machen, analog zu E8. Notwendig ist dies jedoch nicht. Nach zehnmaliger Berechnung (mit B3=1 natürlich, sonst funktioniert das Auszählen in der Hilfsspalte ja nicht) sieht das Spreadsheet dann wie unten abgebildet aus.

S u p e r C a l c : R a n g l i s t e m i t S C e r s t e l l e n

Einen Haken hat die Sache noch: Sind zwei Datenwerte gleich, so erscheinen sie in der sortierten Spalte nur **einmal**. Dies ist nicht unbedingt falsch, aber doch zuerst verwirrend.

	A	B	C	D	E	F
1	Rangliste		RANGDEMO col man			
2						
3	Schalter	1				
4	Zähler	10				
5						
6		Daten	Hilf	Rang	Sort	
7		=====				
8	1	4.54 !	0	6	7.8	
9	2	5.62 !	0	2	5.62	
10	3	5.1 !	0	3	5.1	
11	4	4.84 !	0	5	5	
12	5	5 !	0	4	4.84	
13	6	4.38 !	0	9	4.54	
14	7	4.5 !	0	7	4.5	
15	8	7.8 !	0	1	4.44	
16	9	4.44 !	0	8	4.38	
17	10	4.02 !	4.02	10	4.02	
18		=====				
19		Maximum:	4.02	10		

Es erhebt sich jetzt natürlich sofort die Frage, ob man die Sortierung nicht auch ohne die Rangliste bekommen kann. Auch das geht! Wir lassen die Spalten A, B und C so wie eben. In D bauen wir die sortierte Liste auf, E wird uns dabei wieder als Hilfsspalte zum Auszählen nützlich sein. Das Prinzip ist das gleiche wie eben. In E8 tragen wir ein: IF(B3>0, IF(B8=E19,0,E8), B8)

Die unterstrichene Null ist bei negativen Daten wieder anzupassen (s. o.). In E19 wird das Maximum gebildet: MAX(E8:E17)

In der D Spalte tragen wir analoge Formeln zu eben ein. In D8 also: IF(B4=A8,E19,D8)

Die weiteren Formeln können wieder mit /R erzeugt werden, wobei jedoch nur die Zellen in Zeile 8 angepaßt werden dürfen, also wieder mit Option A arbeiten. Wer will, kann die D Formeln wieder vom Schalter abhängig machen, notwendig ist dies nicht. Am Schluß der Berechnung sieht das Spreadsheet dann so aus:

	A	B	C	D	E	F
1	Sortieren - eine Spalte		SORTDEM1 col man nonext			
2						
3	Schalter	1				
4	Zähler	10				
5						
6		Daten	Sort	Hilf		
7		=====				
8	1	4.54 !	7.8	0		
9	2	5.62 !	5.62	0		
10	3	5.1 !	5.1	0		
11	4	4.84 !	5	0		
12	5	5 !	4.84	0		
13	6	4.38 !	4.54	0		
14	7	4.5 !	4.5	0		
15	8	7.8 !	4.44	0		
16	9	4.44 !	4.38	0		
17	10	4.02 !	4.02	0		
18		=====				
19		Maximum:		0		

C P / M: XDIR

XDIR und die Angaben über Dateien etc.

(Herbert zur Nedden, 2000)

Lasse ich XDIR auf eine Diskette oder RAM-Floppy los, kann das Resultat etwa wie folgt aussehen (ich habe die Dateinamen nicht wegen der Geheimhaltung nicht ins Info aufgenommen, sondern, weil sie lediglich unnötig Platz fressen):

A>xdir h:

XDIR III, Version 1.5 Vertical Listing by File Type and Name
 Disk: H User: 0, File Attributes: Non-System

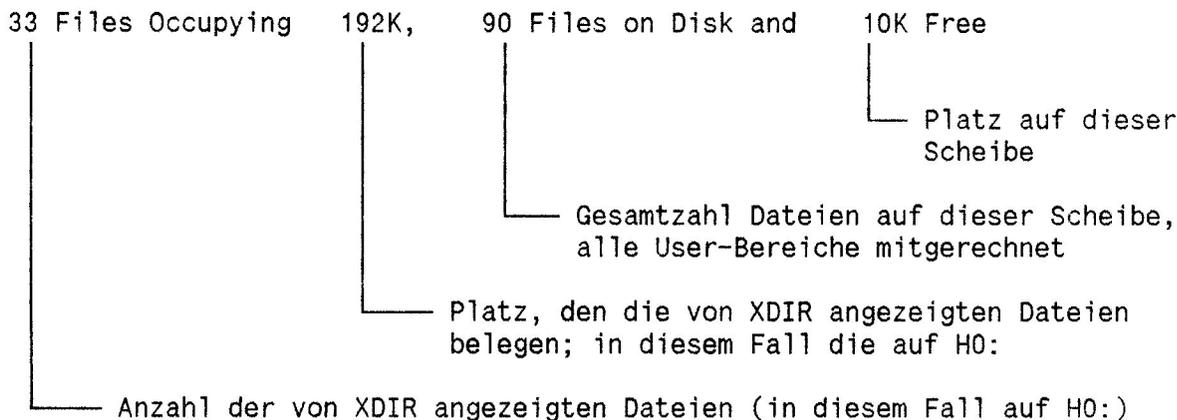
Filename.Type	Size	K	RS	Filename.Type	Size	K	RS	Filename.Type	Size	K	RS
---------------	------	---	----	---------------	------	---	----	---------------	------	---	----

hier stehen diverse Dateien

33 Files Occupying 192K, 90 Files on Disk and 10K Free

Folglich behauptet XDIR, auf H0: seinen 33 Dateien, die 192k belegen. Tja, dann steht da aber noch etwas von 90 Dateien und lumpigen 10k, die noch frei wären. Das kann schon verwirren - jedenfalls so Hartmut Traber, der mich um Aufklärung bat. Ich muß zugeben, daß mir diese Angaben noch nie auffielen. Kurzerhand nahm ich DiJey zur Hand, und schaute mal auf H0: 33 Dateien, die 192k Belegen. Soweit so gut. Nun mal H?: (H*: geht auch, das heißt alle Userbereiche von H:) mit DiJey eingeloggt: Tatsache - 90 Dateien auf der Scheibe - und daß noch 10k Frei seien, war nun auch glaubwürdig. Fazit:

Diese verwirrende Angabe ist völlig O.K.:



A s s e m b l e r: T i p s**Assembler - nicht der Kurs**

(Herbert zur Nedden, 2000)

Kürzlich telefonierte ich mit Peter Lang. Er äußerte die Vermutung, daß der Assembler-Kurs gestorben sei. "Nun ja ...", meinte ich, "er ist nur scheinot". Kurt-Bernd Rohloff hat mir gesagt, es käme noch etwas. In diesem Zusammenhang deutete Peter an, daß er eigentlich gerne mal ... nur wüßte er nicht so recht wie! Auch ist nicht ganz klar, wie COPYD seine Puffergröße ermittelt und überhaupt.

Daher möchte ich hier den einen oder anderen kleinen Tip loswerden - und falls die Resonanz entsprechend ist, steht einer kleinen Serie von Programmier-Tips für Assembler unter CP/M und RAM nichts im Wege außer einer eventuellen Ideen-Flaute bei dem, der eben diese für diese Serie liefert.

Hier erst einmal das typischste Anfängerprogramm der Programmierkurs-Welt:

```
.Z80                diese Zeile braucht der M80
ASEG                diese Zeile möchte der M80 auch gerne haben
org      100h

ld      de,Text      ; Adresse des auszugebenden Strings
                        ; ins DE-Register laden
ld      c,9           ; BDOS-Funktion 9 = String-Ausgabe
call    5             ; BDOS aufrufen
ret                                ; zurück nach Hause

; Hier steht der auszugebene Text - wenn auch nicht der Klassiker
; 'Hello World', aber auch dieser tut es.
; Beachte: BDOS erkennt das Ende des Strings am '$'
Text:   db      10,13,'Nothing happens','$'

end                auch diese Zeile will der M80 haben
```

Nun kurz durch den Assembler gejagt, was ich hier am Beispiel des M80 aufzeige:

Das o.g. Programm (naja, eigentlich ist's ein Promilligramm) speichern wir als Datei mit dem Namen XYZZY.MAC - falls mit NewWord geschehen: NonDocument!

```
A>M80 =XYZZY      assemblieren
A>L80 XYZZY,XYZZY/N/E  linken
A>XYZZY          aufrufen
```

Diese drei Befehle sollten genügen.

Allerdings wird diese Art der Stringausgabe kaum verwendet - zumindest nicht auf unserem Rechner, weil sie nur bedingt KLICK-tauglich ist.

Es geht auch so (und auch schneller):

```
.Z80                diese Zeile braucht der M80
ASEG                diese Zeile möchte der M80 auch gerne haben
org      100h

ld      hl,Text      ; Adresse des auszugebenden Strings
                        ; ins HL-Register laden
call    PrtStr       ; String-Ausgabe aufrufen
ret                                ; zurück nach Hause

weiter geht's auf der nächsten Seite!
```

A s s e m b l e r: Tips

```

PrtStr: ld    a,(h1)      ; Zeichen holen
        or    a          ; ist es Null
        ret   z          ; ja, dann zurück, da fertig
        inc  h1         ; nicht vergessen Zeiger auf String
                          ; zu erhöhen
        ld   c,a        ; Zeichen ins C-Register!!
        push h1         ; Zeiger retten (unter RAM 4.x nicht nötig)
        call Offd3h     ; Zeichen ausgeben über den Systemeinsprung
        pop  h1         ; Zeiger wieder zurückholen (falls gerettet)
        jp   PrtStr     ; Weiter mit dem nächsten Zeichen

        ; Hier steht der auszugebene Text - wenn auch nicht der Klassiker
        ; 'Hello World', aber auch dieser tut es.
        ; Beachte: Ende wird durch eine Hex-0 angegeben
Text:   db    10,13,'Nothing happens',0

        end                auch diese Zeile will der M80 haben

```

Stack und Platz unter C/PM:

Gelegentlich findet man in Programmen die folgende Sequenz:

```

ld    hl,(6)
ld    sp,hl

```

oder andere Variationen der Verwendung des Wertes, der an Adresse 6 steht, um Obergrenzen einiger Bereiche zu ermitteln. Ganz heiße Programme nehmen allerdings an Stelle des Wertes aus Adresse 6 den, der an Adresse 1 steht.

Was hat es wohl damit auf sich ?

Ganz einfach: Der Programmierer kennt die Z80 nicht oder will mit Gewalt die alte 8080 unterstützen. Die Z80 kennt's auch kürzer:

```

ld    sp,(6)

```

Nun zum Zweck und Hintergrund:

Daß ein JP 0 einen WarmBoot auslöst und ein CALL 5 das BDOS aufruft ist vielleicht noch irgendwo im Hinterstübchen zu finden - spätestens jetzt, nachdem Du diesen Satz gelesen hast! An den Adressen 0 und 5 stehen Jump-Befehle auf die entsprechenden Routinen, nämlich auf den zweiten BIOS-Einsprung bzw. auf den BDOS-Einsprung.

Das BIOS beginnt wie folgt:

```

BIOS:
        JP    ColdBoot
WBoot: JP    WarmBoot
        JP    .... u.s.w.

```

D.h. das BIOS beginnt mit einer Sprung-Leiste, deren erster Einsprung auf die BIOS-Funktion 0, den ColdBoot, der zweite auf Funktion 1 (WarmBoot), der dritte auf Funktion 2 (Console-Status), ... zeigt. Der JP an Adresse 0 springt an den Label WBoot, d.h. auf die Adresse BIOS +3.

A s s e m b l e r: T i p s

Will ich die BIOS-Funktion Nummer 5 (Drucker-Ausgabe) aufrufen, kann ich das z.B. wie folgt tun (das Zeichen steht schon im C-Register):

```

LD    HL,(1)           ; Lade Adresse BIOS +3
LD    DE,5             ; Funktion 5
DEC   DE               ; -1, da BIOS +3 = Funktion 1
ADD   HL,DE
ADD   HL,DE
ADD   HL,DE           ; +3x (Funktion -1),
                       ; da ein JP drei Bytes lang ist
JP    (HL)            ; Aufi gehts
    
```

Der Einsprung und Aufruf des BDOS geht ja etwas anders von Statten. Beim BIOS muß ich aus meinem Programm heraus die einzelnen Einsprünge suchen, während das BDOS nur einen Einsprung hat, und die Funktionsnummer im C-Register mitgegeben bekommt. Daher sieht der Aufruf des BDOS so aus:

```

LD    C,Funktionsnummer
CALL  5
    
```

Ein Beispiel dieses Aufrufs steht weiter oben: Version 1 von XYZZY.

Das BDOS beginnt übrigens so:

```

BDOS:
      DS      6           ; Sechs Bytes Serien-Nummer o.ä.
Entry: JP    Los
      ...
    
```

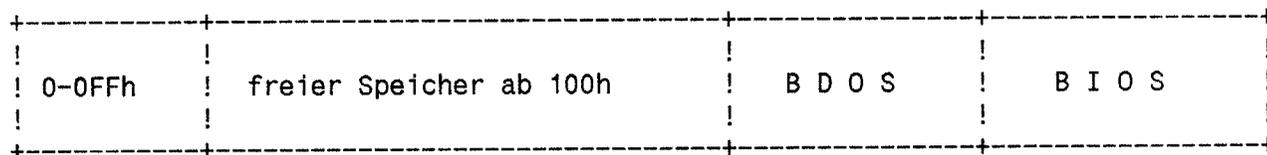
An Adresse 5 steht ein Jump auf Entry, also auf BDOS +6.

Zurück zu der Frage, warum Programme gelegentlich folgendes beinhalten:

```

ld    hl,(6)
ld    sp,hl
    
```

Hier die Speicheraufteilung unter CP/M:



Ein Programm, welches wie mein COPYD das BDOS nicht benötigt, darf sich den gesamten Speicher von 100h bis zum BIOS schnappen. Die meisten Programme brauchen hingegen auch das BDOS - z.B. für Disketten-Zugriffe, und müssen sich daher mit dem Speicher ab 100h bis zum BDOS begnügen. Das ist aber so wenig nicht, wie z.B. NewWord beweist!

Da jedoch die Startadresse des BDOS und des BIOS nicht 100%-ig fest liegen, sondern von der Systemgröße, die mit SYSGEN (CP/M) bzw. WRITESYS (RAM) festgelegt werden, abhängig sind, können diese Werte nicht einfach als Konstanten im Programm angegeben werden.

A s s e m b l e r: T i p s

Also so Dinger wie

```
BDOS    equ    0D300h
BIOS    equ    0E100h
```

sind absolut unfein - sie stimmen nämlich nur für ein 58k-CP/M.

Also schauen Programme nach, wo BDOS und BIOS beginnen: an Adresse 6 und 1. Wenn nämlich an Adresse 0 ein Jump auf BIOS+3, d.h. den WarmBoot steht, steht an Adresse 2 der Wert BIOS+3. Für den BDOS-Einsprung gilt nämliches.

Um die Platz-Grenze zu ermitteln:

1. Wenn das Programm das BDOS nicht braucht:

```
ld      hl,(1)                ; Lade BIOS +3
dec     hl
dec     hl
dec     hl
dec     hl
ld      (TopMem),hl          ; BIOS-Beginn -1 = letztes freies Byte
```

2. Wenn das Programm das BDOS nicht braucht:

```
ld      hl,(6)                ; Lade BDOS +6
dec     hl
ld      (TopMem),hl          ; BDOS-Beginn +5 = letztes freies Byte
```

Das genügt, da keiner die Seriennummer des BDOS benötigt, diese also locker überschrieben werden kann.

3. Anständige Programme hingegen, die das BDOS brauchen, lassen auch die BDOS-Seriennummer unverändert, d.h. nehmen z.B. diese Variante:

```
ld      hl,(6)                ; Lade BDOS +6
dec     hl
ld      (TopMem),hl          ; BDOS-Beginn -1
```

Bzw.

```
ld      hl,(6)                ; Lade BDOS +6
ld      de,0-7
add     hl,de
ld      (TopMem),hl          ; BDOS-Beginn -1
```

Wichtig: Überschreibt das Programm das BDOS und/oder den davor liegenden CCP (den ich hier einfach außer acht gelassen habe, da jedes größere Programm diesen rauswirft), muß das Programm mit einem WarmBoot (JP 0) enden, damit CCP und BDOS nachgeladen werden.

A s s e m b l e r: M a c r o s

Daß Programme nun nicht (wie anscheinend richtig)

```
ld    hl,(6)
dec   hl
ld    sp,hl
```

einfach

```
ld    sp,(6)
```

machen dürfen, liegt daran, daß der Stack-Pointer sp immer hinter das letzte Byte des Stacks zeigt, d.h.

```
push  hl
```

macht:

```
dec   sp
ld    (sp),h
dec   sp
ld    (sp),l
```

oder anders ausgedrückt.

```
dec   sp
dec   sp
ld    (sp),hl
```

Hoffentlich hilft Dir dieser Artikel etwas. Was soll ich fürs nächste Info zum Thema Assembler schreiben ?

Noch zwei MACROs für MACRO-80

(Olaf Krumnow, 2050)

Ich bekomme doch recht häufig Assemblerprogramme von anderen Clubmitgliedern zu sehen und wundere mich doch immer wieder, wie selten MACROs benutzt werden. Dabei sind diese Dinge sehr viel fähiger, als gemeinhin angenommen. Darum will ich hier noch zwei Beispiele bringen, um zu zeigen was alles möglich ist. Das erste Beispiel ist noch recht einfach. Es ist ein Macro, das das Verschieben mittels LDIR vereinfacht, indem die Register gesetzt werden. Nebenbei optimiert dieses Macro, falls die Anzahl zu verschiebender Bytes kleiner 2 ist. Das zweite Macro ist da schon etwas komplexer. Es programmiert die Multiplikation von HL mit einer Konstante mittels einfacher Additionen.

A s s e m b l e r: Macros

Doch fangen wir mit MOVE an:

```

00:move      macro   von,nach,anz
01:    ife NUL von
02:        ld      hl,von
03:    endif
04:    ife NUL nach
05:        ld      de,nach
06:    endif
07:    if NUL anz
08:        ldir
09:    else
10:        if anz gt 2
11:            ld      bc,anz
12:            ldir
13:        else
14:            ldi
15:            if anz eq 2
16:                ldi
17:            endif
19:        endif
20:    endif
21:        endm

```

Die Zeilennummern werden natürlich nicht mit eingegeben. Sie dienen nur der einfacheren Referenz im Text.

Zeile 00 definiert den Kopf des Macros mit den drei Parametern. Diese Parameter beschreiben die Quelladresse (von), die Zieladresse (nach) und die Anzahl (anz) zu verschiebender Bytes. Alle diese Parameter können im Aufruf weggelassen werden, falls das Programm die Werte zuvor berechnet. In den Zeilen 01-06 werden Quell- und Zieladresse dann in die entsprechenden Register eingetragen. Falls keine Anzahl angegeben ist, wird ein LDIR (07-09) ausgeführt. Ansonsten wird geprüft, wieviele Bytes verschoben werden sollen. Ab drei Bytes (10) wird das BC-Register mit der Anzahl geladen (11) und ein LDIR ausgeführt (12). Das ist die übliche Methode. Bei einer Anzahl kleiner als drei ist jedoch die Variante mit ein oder zwei LDI der kürzere und schnellere Weg (13-16). Anschließend werden nur noch alle IFs geschlossen und das Macro beendet (17-21). Das war's dann schon.

Noch ein paar Beispiele:

```

move  start,ziel,5      ergibt
    ld  hl,start
    ld  de,ziel
    ld  bc,5
    ldir

move                                     ergibt
    ldir

move  start,,2          ergibt
    ld  hl,start
    ldi
    ldi

```

A s s e m b l e r: M a c r o s

Kommen wir zum zweiten Macro. Das ist dann schon etwas komplexer.

```

00:mult      macro   k
01:...l      aset   k
02:      if ...l eq 0
03:          ld     hl,0
04:          exitm
05:      endif
06:      if ...l eq 1
07:          exitm
08:      endif
09:...n      aset   0
10:      rept   16
11:      if (...l gt 1) and ((...l mod 2) ne 0)
12:          push  hl
13:...n      aset   ...n+1
14:      endif
15:...l      aset   ...l/2
16:      if ...l eq 0
17:          exitm
18:      endif
19:          add   hl,hl
20:      endm
21:      rept   ...n
22:          pop   de
23:          add   hl,de
24:      endm
25:endm

```

Wie gesagt, dieses Macro multipliziert das HL-Register mit einer Konstante. Der Hintergrund ist, daß sich eine Konstante in eine Summe von Zweier-Potenzen zerlegen läßt, so ist z.B. $5 = 2^2 + 2^0$. Und Zweierpotenzen lassen sich recht einfach durch verschieben nach links berechnen (ADD HL,HL). So muß das Macro also nur die richtigen Zweierpotenzen berechnen und sie zum Schluß alle addieren. Dieses Verfahren ist üblicherweise sehr verbreitet und wird häufig benutzt, nur scheint es jeder für jede Konstante neu zu kodieren. Dabei macht es dieses Macro doch viel einfacher (insbesondere, wenn eine Konstante mal geändert werden muß). Zeile 00 definiert wieder das Macro und seinen Parameter (die Konstante). In Zeile 01 wird dieser Wert in eine lokale Variable kopiert (ich habe die beiden benutzten Variablen mit drei führenden Punkten definiert. Das sieht zwar etwas merkwürdig aus, aber ich verhindere damit Kollisionen mit im Programm definierten Labels). Dann wird getestet, ob die Konstante 0 ist (02). In diesem Fall muß natürlich auch 0 als Ergebnis geliefert werden (03). Danach wird das Macro abgebrochen (04). Dann die Abfrage auf 1. In dem Fall bleibt HL natürlich unverändert (06-08). Und dann gehts richtig los: In Zeile 09 wird ein Zaehler initialisiert. Zeile 10 startet ein Zähl-Macro. Die 16 steht für die max. 16 Bits, die die Konstante lang sein kann. Zeile 11 enthält dann die wichtigste Abfrage: Ist die momentane Zweierpotenz in der Konstante enthalten und gibt es noch eine größere? Dann muß ihr Wert nämlich auf den Stack gerettet werden, um sie später zum Ergebnis addieren zu können. Ist das der Fall, so steht in Zeile 12 das PUSH, in Zeile 13 wird der Zähler erhöht. Ihn brauchen wir nachher, um die richtige Anzahl von Werten wieder vom Stack zu holen. Dann wird die Konstante ein Bit nach rechts geschoben (= Division durch 2). Ist sie jetzt null, so sind wir mit dem ersten Teil fertig (16-18). Ansonsten wird HL mit zwei multipliziert (19) und das ganze wiederholt (Das ENDM in Zeile 20 gehört zum REPT in Zeile 10!!). In den Zeilen 21-24 werden die Summanden wieder vom Stack geholt und auf das Ergebnis aufaddiert. Et voila.

K o m i k: Noch'n Gedicht

Beispiele:

```
mult 0          ergibt
  ld  hl,0
```

```
mult 2          ergibt
  add hl,hl
```

```
mult 5          ergibt
  push hl
  add hl,hl
  add hl,hl
  pop  de
  add hl,de
```

Die Beispiele zeigen doch recht deutlich, daß Macros sehr leistungsfähig sind. Bleibt nur zu hoffen, daß auch andere Clubmitglieder ihre Macros mal veröffentlichen. Ich bin doch wohl nicht der einzige, der diese Dinger nutzt, oder?? Ach nein, Herbert hat ja auch schon mal was zu dem Thema geschrieben...

Hier zwei Gedichte

1. Das Festplatten-Liedchen (von geplagten DoMessDos-Usern):

```
"Zehn kleine MegaBytes
war'n froh und frei geblieben,
kam ein neues Update an,
da waren's nur noch sieben."
```

(Aus "TOOLBOX" 10/89)

2. Zum Thema Ventilatoren könnte Wilhelm Busch, so er von solchen geplagt seine Gedichte verfaßt hätte, evtl. folgendes geschrieben haben:

```
"Ach, was muß man oft von bösen
Rechnern hören oder lesen;
störend werden sie empfunden,
weil meist sie mit Geräusch verbunden."
```

(Aus c't 10/89)

D RAM's - ab LAGER
 256KX1, 70, 100, 120, 150 Nanu-
 sek., diverse Hersteller, Tel. [REDACTED]
 [REDACTED], FAX [REDACTED] 

← *Nanu*

Hardware: c't-SRAM-Floppy**c't-SRAM-Floppy: Probleme**

(Herbert zur Nedden, 2000)

Unlängst berichtete ich im Info, daß die Statik-RAM-Floppy von c't gelegentlich dazu neigt, nach einem Schreibzugriff jegliche Lesezugriffe zu verweigern, bis der Rechner aus- und wieder eingeschaltet wird. Das Drücken von Reset genügt nicht. Da ich Olaf Krumnow's Rechner mal wieder in den Fingern hatte, habe ich mich wieder mal auf die Suche gemacht, da dieser Rechner je nach dem, welche Karten auf dem Bus steckten Ärger machte, oder nicht.

Ist im FDX-Bus die Interface-Karte zur Hauptplatine, die 80-Zeichen-Karte sowie mittels zweier FDX-ECB-Adapter von Gerhard Witzel die Edicta-Grafikkarte und eben diese SRAM-Floppy, so läuft alles bestens (bis auf die Diskettenzugriffe, die bekanntlich ohne Floppy-Controller nicht laufen). Sobald jedoch der Floppy-Controller hinzugesteckt wird, machte die SRAM-Floppy Ärger: Schreibzugriffe werden 100%-ig korrekt ausgeführt. Anschließende Leseversuche scheitern jedoch kläglich!

Mein Oskar bewies mir, daß diverse Signale, wie insbesondere /IOREQ (der / bedeutet 'Quer-Balken'), /RD und /WR von schlecht zu beschissen übergehen, sobald der Floppy-Controller mit von der Partie ist. Vorher ist der High-Pegel dieser Signale ein Wust zwischen 4 und 5 Volt. Nachher kommen noch kurze Einbrüche unter 3.5 Volt hinzu. Versuche, diese Signale mittels eines anderen Treibers auf der Interface-Karte (es ist der 74LS241) zu verbessern nutzen kaum. Die unschönen Einbrüche in den Signalen, überleben weiterhin.

Obwohl diese Signale so bescheiden aussehen, läuft die Kiste bestens, solange ich nicht auf die SRAM-Floppy schreibe. Und selbst ein Schreibzugriff läuft bombig - nur ist anschließend Lesen von dieser Floppy unmöglich. Magisch!

Wie funktioniert der Zugriff auf die SRAM-Floppy?

Auf der SRAM-Floppy ist ein Zähler (74LS393), der auf Null gesetzt wird, sobald ich der Karte den zu bearbeitenden Sektor mitteile. Anschließend wird dieser Zähler nach jedem Datenzugriff um Eins erhöht. D.h., daß ich einfach Sektor und Spur an die SRAM-Floppy übermittle, und anschließend 128 Datenzugriffe mache, um einen 128-Byte langen Sektor zu lesen bzw. zu schreiben.

Habe ich jedoch unsaubere Signale mit diesen unschönen Einbrüchen, erhält die Karte gelegentlich bei einem Datenzugriff nicht nur dieses Signal, sondern zwischendrin auch noch so einen Einbruch, der ebenfalls den Zähler erhöht. Und schon ist ein Byte übersprungen. Grand Malheur de Kack!

Betrachte ich (rein philosophisch) einen solchen Einbruch im Signal mal statt von Masse aus von 5 Volt aus an, so ist es ein Peak, d.h. genau daß, was man mit einem kleinen Kondensator killen kann. Flugs einen 220pF-Kondensator zwischen 5 Volt und das Signal 'erhöhe Zähler' gelötet, und - oh Wunder - die Chose läuft! (Kondensator zwischen 74LS393, Pin 1 und den daneben sitzenden 74LS04, Pin 14.)

Diese Lösung ist keineswegs sauber, da ich statt der Ursache nur das Symptom beseitige. Da aber die Karte so funktioniert, soll's halt so bleiben.

Hallo Bastler und Tüftler, was sagt Ihr zu dieser Problematik? Ich weiß immer noch nicht, warum diese Probleme erst nach einem Schreibzugriff auftreten! Gutgemeinte Zuschriften bitte an Herbert zur Nedden, Sonnenau 2, 2000 Hamburg 76 oder telefonische Kontaktaufnahme unter (040) 200 87 04. Danke!

H a r d w a r e: 80-Zeichenkarte und höhere Geschwindigkeit

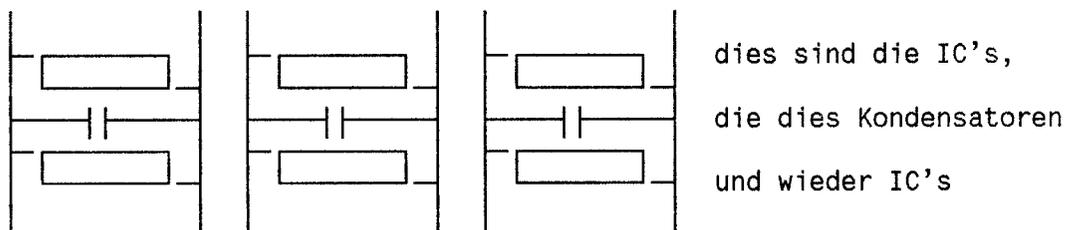
80-Zeichenkarte unter 8 MegaHertz

(Herbert zur Nedden, 2000)

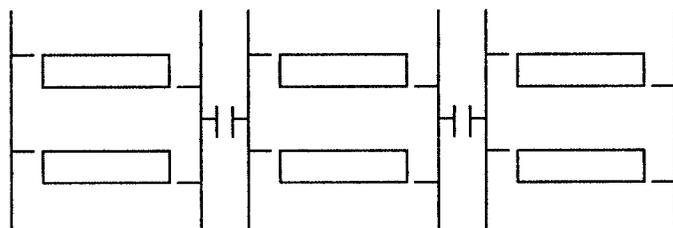
Damit die 80-Zeichenkarte überhaupt unter 8 MHz tut, muß vom IC 1C (74LS74) der Pin 5 über der Platine abgekniffen, und unter der Platine Pin 8 mit dem ehemaligen Anschluß von Pin 5 dieses IC's verbunden werden. Aber das ist ja alles alter Käse, d.h. stand mehrfach im Info.

Doch trotzdem stehen einige Zeichen unter 8 MHz gelegentlich nicht da, wo sie hin gehören oder haben ein falsches Attribut. Horst Kupka konnte die Betriebsprobleme durch einen Tantal-Kondensator, den er über das Attribut-RAM (d.h. zwischen Masse und 5 Volt) gelötet hat, beheben. Aber auch dies, sowie die Verbesserung der Kühlung dieser Karte brachte nicht das gewünschte Resultat. Olaf Krumnow beklagte sich trotzdem über Zeichen, die unter 8 MHz nicht da erschienen, wo sie hingehörten. Allerdings erst nach längerer Arbeitszeit, d.h. wenn der Rechner warm geworden war. Immerhin trat dieses Problem ohne Kühlung und Tantal-Kondensator (1 uF) erheblich früher auf.

Als ich im Rahmen meiner u.g. Umbauarbeiten die 80-Zeichenkarte genauer angesehen habe, stellte sich heraus, daß diese Karte einen erheblichen Designfehler hat! Normalerweise werden auf Computer-Platinen Stützkondensatoren vorgesehen, die zwischen Masse und 5 Volt stecken, und zwar so, daß sie dicht an den Anschlüssen der IC's sitzen. D.h. das Bild müßte etwa so aussehen:



Memotech hat sich die Arbeit erheblich einfacher gemacht:



Offensichtlich ist es so viel leichter, die Kondensatoren unterzubringen - nur daß sie dort fast keine Wirkung entfalten! Die Kondensatoren sollen schließlich als 'kleine Batterie' die Stromversorgung der IC's unterstützen, was klar erfordert, daß sie auch an der richtigen Stelle (oberes Bild) sitzen.

Also habe ich auf Olaf's Karte ca. ein Dutzend Kondensatoren zu 100 nF direkt auf verschiedene IC's gelötet, d.h. einen Pin des Kondensators an den Masseanschluß und den anderen an den 5 Volt-Anschluß des IC's. Seit dem läuft die Karte auch unter 8 MHz soweit korrekt. Unter dem Oszilloskop sieht die 5 Volt-Stromversorgung obendrein nun erheblich sauberer aus!

- F: Auf welche IC's sollen die Kondensatoren ?
- A: Einfach gleichmäßig über die Platine verteilt.

Hardware: 80-Zeichenkarte und Grafik

80-Zeichenkarte als Hercules

(Herbert zur Nedden, 2000)

Als mich meine Lötlust mal wieder überkam, habe ich mich drangesetzt, um meine 80-Zeichenkarte auf die von Horst Kupka entwickelte Grafikfähigkeit zu erweitern. Erfreulicherweise hat sich Holger Göbel schon einmal diese Mühe gemacht, und dabei die Dokumentation von Horst verifiziert (schönes neudeutsches Wort für überprüft), wie in Info 32 zu sehen war.

Die in der Umbauanleitung genannten Pin-Abkniefereien habe ich noch recht leicht geschafft - allerdings habe ich die ROM-Pins 18 nicht hochgebogen, sondern die Leitungen auf der Platine gekappt. Beim Grafik-EPROM muß dies zu beiden Seiten des Pins 18 geschehen, einmal neben und einmal 'im' Sockel auf der Platinenoberseite. Diese so aufgetrennte Leiterbahn mußte mit einem vernünftigen Draht nachverdrahtet werden (da es eine Masse-Leitung ist). ACHTUNG: Die breite Leiterbahn am Rand auf der Platinenunterseite ist eine +5 Volt-Leitung, keine Masse-Leitung! Beim Zeichensatz-EPROM (sitzt weiter innen auf der Platine) genügte ein Kratzer auf der Platine. Damit habe ich die Möglichkeit, die EPROMs ggf. mal ohne LötKolben auszutauschen, da ich das neue Signal für in 18 nun unten an den in des IC-Sockels löten konnte.

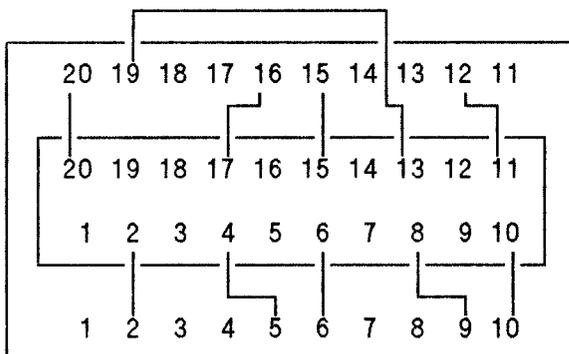
Allerdings habe ich mich mit dem Vorschlag, einige der IC's auf eine kleine eigene Platine zu setzen überhaupt nicht anfreunden können!!! Das ist mir zu frikkelig und unübersichtlich - zumal nicht nötig. Außerdem sind die IC's recht wenig untereinander, sondern reichlich mit anderen auf der Karte verbunden. Daher landeten alle N?-IC's, d.h. die neuen, auf alten der 80-Zeichenkarte, und zwar wie folgt:

N1 (74LS175) auf 5C, weil hier die Signale schon zur Hand sind.
 Damit es sich lohnt, wird er nicht mittig oder wenigstens voll auf seinen größeren Unterbau gesetzt, sondern so, daß Pins 1 und 16 des N1 überstehen, da so dessen Eingänge an Pins 4, 5, 12 und 13 direkt auf den Pins 3, 4, 17 und 18 des 5C sitzen, wo zufällig die Signale D4-D7 anliegen. Die anderen Pins hochbiegen und entsprechend verbinden. Aber Achtung: Die Belegung des N1 ist neu:

Signal:	HGR	$\overline{\text{HGR}}$	TXTPG	ROM1	$\overline{\text{ROM1}}$	ROM2	$\overline{\text{ROM2}}$
Pin:	10	11	9	2	3	15	14

N2 (74LS157) auf 3A, nur Pin's 8 und 14 mit dem unteren verbunden!
 Schließlich sind seine Anschlüsse an den 4A (CRTC) und 3B' zu finden, die so angenehm dicht bei sitzen.

N3 (74LS244) auf 9B (das ist ein LS273)
 Hier kommt reichlich Lötvereinfachung ins Spiel, da 9B acht der Informationen, die in N3 hinein müssen liefert, und siebene direkt vom unteren Pin bzw. dessen Nachbarn abgegriffen werden können, sowie Masse und 5 Volt auch günstigerweise passen:



Großer Kasten:
 unteres IC: 9B (72LS273)
 Kleiner Kasten:
 oberes IC: N3 (72LS244)

Hardware: 80-Zeichenkarte und Grafik

D.h. Pins 1, 2, 6, 15 und 20 werden senkrecht an den Untermann gelötet, Pins 4, 8, 11 und 17 müssen schräg gebogen werden, um an den um eins versetzten Pin des Untermanns zu kommen, Pin 13 muß an Pin 19 des Unteren IC's verkabelt werden. Die restlichen Pins mal waaregecht biegen.

Also: Signal lt. Schaltplan	S0	S1	S2	S3	S4	S5	S6	S7
Pin 9B (unten, LS273)	2	5	6	9	12	15	19	16
Pin N3 (oben, LS244)	2	4	6	8	11	15	13	17

Und: Signal lt. Schaltplan	P0	P1	P2	P3	P4	P5	P6	P7
Pin N3 (oben, LS244)	18	16	14	12	9	5	7	3

N4 (74LS166) auf 7E (das ist der LS157 mit den drei neuen 1k-Widerständen)
Auch hier nur Pins 8 und 16 nach unten verbinden. Die meisten seiner Leitungen werden an 8E bzw. 3D angeschlossen (siehe Schaltplan).

N5 (72LS138) auf 2C, wie in der Anleitung sinnvoll vorgeschlagen.
Hinweis: Im Schaltplan ist eine Unstimmigkeit mit der Beschreibung. Pin 5 des N5 kann wahlweise (Anleitung) an den darunter sitzenden LS138, oder (Schaltplan) an Pin 4 des N5. Ich wählte erstere Lösung, d.h. die laut Umbauanleitung, da die besser ist.

N6 (74LS374) auf 4C, wie empfohlen.

N7 (74LS08) auf 2B (ich glaube das ist ein LS30, nur sehen kann ich's nimmer)
Dabei nur Pin's 8 und 14 mit dem unteren verbunden!

Fehler im Schaltplan von Horst Kupka:

IC 9E: Pins 1 und 2 vertauschen.

Diesen Fehler im Schaltplan entdeckte ich, da ich vom Pin 1 des 9E ein Signal für N4 (74LS166) abgreifen wollte, jedoch beim Nachmessen gemäß Umbauanleitung feststellen mußte, daß ich nicht das Signal erwischte hatte, welches ich brauchte. Kurzen Nachmessen ergab die o.g. Entdeckung.

Fehler im Schaltplan von H. Bleekermoelen:

IC 9B: richtige Pin-Signal-Zuordnung siehe den Plan von H. Kupka
(Dieser Fehler war schon in Memotech's Schaltlänen!)

Löten dieser Karte

Da ich mir aufschreiben wollte, welche Drähte ich legen muß, um der Karte Dampf zu machen, habe ich mir gedacht, vielleicht hilft's Dir, wenn ich's für Info aufschreibe.

Um mir die Arbeit zu vereinfachen, haben die neuen IC's N1 bis N7 je einen weiteren Namen erhalten, nämlich einen, aus dem ich erkennen kann, wo das Teil sitzt, als da folglich sind:

3B': 74LS367 = N0, der von der Wenzek-Aufrüstung her stammt.

Die Bredereke-Variante habe ich nicht beschrieben, siehe ggf. Schaltplan

5C': 74LS175 = N1 2C': 74LS138 = N5

3A': 74LS157 = N2 4C': 74LS374 = N6

9B': 74LS244 = N3 2B': 74LS08 = N7

7E': 74LS166 = N4

H a r d w a r e: 80-Zeichenkarte und Grafik

Zur Notation: 2B/4 heißt IC 2B, Pin 4
 2B/4 - 5C/6 heißt Kabel von 2B/4 nach 5C/6
 2B/4 - 5C/6 - 7A/10 heißt Kabel von 2B/4 über 5C/6 nach 7A/10

Bevor Ihr mit dem Verkabeln gemäß meiner unten folgenden Draht-Liste beginnen könnt, müssen folgende Vorbereitungen erledigt sein:

1. IC 3B' ist ein 74LS367 gem. H. Wenzek: Pins 1, 8, 15 und 16 mit IC 3B verbunden, 2, 4, 6 und 14 an Masse, restliche Pins hochgebogen.
Pin 2 des IC 3B wurde auf der Platinenunterseite von Masse abgetrennt.
2. Folgende Vorarbeiten sind gemäß Info 32, Seite 31 f. gemacht:
1E/2, 8E/1, 6A/1, 6A/6, 7F/2, 1B/8, 5F/10 und 1C/1 bis 1C/6 sind hochgebogen oder sonst wie von ihrem bisherigen Anschluß abgetrennt.
3. IC's 7A und 9A haben gemäß meiner o.g. Ausführungen je den Pin 18 am Sockel von Masse abgetrennt.
4. Die ICs 7C und 9C (RAMs 61256) wurden gemäß Info 32 vorbereitet: Pins 1, 2, 23, 26 und 28 hochgebogen, RAMs so in alte Sockel gesteckt, Pins 1,2 und 27,28 überstehen, Pin 27 mit Sockel-Pin unter Pin 23 verbunden und Pin 28 an 5 Volt (Sockel-Pin unter Pin 26) angeschlossen.
5. Die drei Widerstände zu 1 kOhm wurden zwischen Masse und Pins 5, 13 und 14 des 7E gelötet (siehe Info 32, Seite 32 kurz unter Seitenmitte)
6. Die IC's N1 bis N7 wurden auf die o.g. ICs 5C, 3A, 9B, 7E, 2C, 4C bzw. 2B gelötet, wobei die Pin-Verbindungen zwischen unterem und oberem IC den o.g. Ausführungen entsprechen; Plus/Masse beim 5C' nicht vergessen!

Tja, bis hierhin ist es noch übersichtlich und relativ simpel, d.h. mit etwas Übung und einem guten spitzen LötKolben locker in 30 Minuten schaffen, falls eine fertige Musterplatine daneben liegt, und man nicht Lesen muß.

WICHTIG: Nun Kontrolle, und zwar sorgfältig und gemütlich!!!!!!

1. Sitzen alle gehuckeackten IC's richtig, sind Masse und 5 Volt angeschlossen ?
2. Insbesondere die IC's 9B und 9B' und die schrägen Pins: Ist hier der richtige in die korrekte Richtung gebogen und am passenden Pin des unteren verlötet - ohne Lötbrücke zu dessen Nachbarn??

----> Später wird es viel Schwieriger, Fehler aus dieser Phase zu beheben!!

Drähte auf der Platinen-Unterseite

Abgesehen von der wegen dem o.g. Vorbereitungspunkt 3 (ROM-Pins) erforderlichen Masse-Leitung und der später noch kommenden Leitung zu den ROM-Pins 18 sind nur zwei Drähte unter der Platine erforderlich:

4A/15	-	3B/2	MA11
6A/6	-	6A/8	CI an Masse

Drähte auf der Platinen-Oberseite

Damit ich überhaupt eine Chance habe, den Kabelwust zu kontrollieren, den ich für die Aufrüstung zusammengelötet habe, habe ich verschiedenfarbige Drähte für die verschiedenen Dinge verwendet.

Hier nun die Verbindungen in der Reihenfolge, in der sie sich vernünftig löten lassen:

Hardware: 80-Zeichenkarte und GrafikZuvor: Masseverbindungen

3A'/15 - 3A'/8
 3B'/14 - 3B'/2 - 3B'/4 - 3B'/6 - 3B'/8
 7E'/1 - 7E'/6 - 7E'/8

Grüne Drähte: Portdekodierung

2B'/5	-	2C/2	A1
2B'/4	-	2C/3	/RD
2B'/6	-	2C'/3	A1 & /RD
2B'/8	-	Anschluß unter 1B/8	/CRTCCS
2B'/9	-	2C'/15	Y0 von LS138
2B'/10	-	2C'/14	Y0 von LS138
2C'/4	-	1B/8	/IO38-3B
2C'/7	-	5C'/9	/OUT3B
2C'/9	-	4C'/11	/OUT3A

Gelbe Drähte: Überbrückung der ROMs

9B/19	-	9B'/13	S.O.
6A/10	-	9B'/9	P4
6A/11	-	9B'/5	P5
6A/12	-	9B'/7	P6
6A/14	-	9B'/3	P7
6A/5	-	9B'/12	P3
6A/4	-	9B'/14	P2
6A/3	-	9B'/16	P1
6A/2	-	9B'/18	P0

Braune Drähte: 2. Schieberegister (74LS166)

7E'/11	-	8E/4	F
7E'/5	-	8E/7	D
7E'/7	-	4F/4	SRCK
7E'/10	-	9E/2	E
7E'/12	-	3D/10	F
7E'/13	-	6A/1	Übertrag
7E'/14	-	3D/7	G
7E'/15	-	5F/8	/SRLD
7E'/2	-	8E/13	A
7E'/3	-	8E/17	B
7E'/4	-	8E/14	C

Blaue Drähte: Adressen

3A'/14	-	4A/17	MA13
3A'/11	-	4A/16	MA12
3A'/13	-	4A/37	Z1
3A'/10	-	4A/38	Z0
3A'/12	-	3B'/12	Z1 oder MA13
3A'/9	-	3B'/10	Z0 oder MA12
3A'/6	-	3B'/11 - 4C'/5	RA13
3A'/7	-	7C/26 - 9C/26	NA13
3B'/9	-	4C/16 - 7C/2 - 9C/2	RA12
3B/3	-	4C/6 - 7C/23 - 9C/23	RA11

(ACHTUNG: 3B, nicht 3B')

H a r d w a r e: 80-Zeichenkarte und GrafikGraue Drähte: Taktgeschichten

1C/1	-	1C/13	
1C/6	-	1C/2	- 3A'/3
8F/5	-	1C/3	- 3A'/2
1C/5	-	2B'/2	
2B'/1	-	5F/4	
2B'/3	-	5F/10	
3A'/4	-	7F/2	

7.5 MHz
15 MHz
/7.5 MHz
CH
einer für /SRLD
Takt (15 oder 7.5 MHz)

Weißer Drähte: Port 3B

5C'/1	-	4A/2	Reset (statt 4A/2: Lötage neben 6A/15)
5C'/7	-	3A'/5	TXTPG
5C'/11	-	9B'/1	- 9B'/19
7E'/9	-	8E/1	- 5C'/10
	-	7C/1	- 9C/1
1E/2	-	1C/4	- 3A'/1
	-	7C/1	
9A/18	-	7A/18	- 9C/1

/HGR
weiter nächste Zeile
HGR
weiter nächste Zeile
HGR (zweites Stück Kabel)
HGR (drittes Stück Kabel)

Bemerkungen zur HGR-Verdratung:

Diese ist in Form von drei Kabelstücken vorgenommen, die jeweils an den Zwischenstellen abisoliert und angelötet wurden. Für den Anschluß an die Pins 18 der ROM's habe ich ein Loch in die Platine als Kabeldurchführung gebohrt.

Nicht verlötete IC-Pins:

3B' = N0 (74LS367): 3, 5, 7, 13
5C' = N1 (74LS175): 2, 3, 6, 14 und 15 (Bedeutung siehe weiter vorne)
3A' = N2 (74LS157): keiner
9B' = N3 (74LS244): keiner
7E' = N4 (74LS166): keiner
2C' = N5 (74LS138): 10, 11, 12, 13
4C' = N6 (74LS374): 2, 6, 9, 12, 15, 16, 19
2B' = N7 (74LS08) : 11, 12, 13

Hardware: Abfrage, ob Diskette gewechselt wurde**Wie kann ich abfragen, ob die Scheibe gewechselt wurde**

(Herbert zur Nedden, 2000)

Es wurde ja im Info angeregt, den Floppy-Controller aufzurüsten und die Laufwerke (soweit möglich) umzujumpen, damit der Cache das DiskChange-Signal abfragen kann, um festzustellen, ob die Diskette wirklich gewechselt wurde - egal was das BDOS behauptet. Soweit so gut. Die Umbauanleitung war ja abgedruckt - und die Anleitung für die umschaltbaren EPSON SD580 ist in diesem Info.

Es kam natürlich die Frage:

'Wie kann ich im Programm das DiskChange-Signal abfragen?'

Um das unten gelistete Programm auch von seiner Funktion her zu glauben oder zu verstehen, muß man wissen, wie die Laufwerke das DiskChange-Signal erzeugen und verwalten.

Wird der Rechner (genauer das Laufwerk) eingeschaltet, d.h. der Strom angeschaltet ODER die Diskette gewechselt, meldet das Laufwerk DiskChange = aktiv. Das Laufwerk setzt sein DiskChange auf nicht-aktiv sobald das Laufwerk bereit war, d.h. sich eine Diskette drin gedreht hat UND das Laufwerk einmal gesteppt, d.h. den Kopf bewegt hat.

Der Cache liest daher das DiskChange-Signal ein und führt vor dem Einlesen des Direktories einen Step in Richtung Diskettenmitte aus. Der Grund hierfür ist, daß bei einigen Formaten das Directory auf eine Spur paßt, und falls der Kopf gerade auf dieser ist erfolgt kein Step. War der Laufwerks-Kopf auf der innersten Spur geht zwar dieser Step in die Hose, dafür Steppt der Kopf dann nach außen, wo das Directory steht.

Da jedoch auch der Cache das DiskChange-Signal unbedingt benötigt, darfst Du ein Programm nicht einfach das DiskChange-Signal nach dem Einlesen mittels Step zurücksetzen! Du mußt den Cache zumindest informieren, daß die Diskette gewechselt wurde. Wenn nicht, merkt zwar Dein Programm, daß die Diskette gewechselt wurde, der Cache jedoch kriegt das nicht mit und liest folglich das Directory der neuen Scheibe nicht ein sondern glaubt, daß sein Puffer noch gültig ist.

Also muß Dein Programm folgende DREI Dinge tun:

1. Einlesen (und merken) des DiskChange-Signals
2. Dem Cache diese Information zuführen
3. Zurücksetzen des DiskChange-Signals

Dem Cache die Information zuzuführen ist kein Problem. Einfach über das BIOS ein Drive-Select ausführen, d.h. das Laufwerk anwählen. Der Cache erhält hierbei nämlich die Kontrolle und merkt dann auch, daß die Diskette gewechselt wurde.

Wurde die Diskette nicht gewechselt, können Schritte 2 und 3 natürlich entfallen, was auch in meinem u.g. Programm der Fall ist.

Selbstmurmelnd muß obendrein zwischen FDX und SDX unterschieden werden, da der Controller an unterschiedlichen Port-Adressen liegt, und bei der SDX die Bits des sog. Control-Byte eine andere Bedeutung als bei der FDX haben.

H a r d w a r e: Abfrage, ob Diskette gewechselt wurde

```

.Z80
ASEG

Start:      ld      a,(80h)          ; Kommando-Zeile
            or      a
            ret     z              ; keine Info dort
            ld      a,(82h)
            ld      a,'C'
            sub     'B'
            ret     c              ; ungültiges Laufwerk
            cp     'F'-'B'
            ret     nc             ; ungültiges Laufwerk
            ld      c,a            ; Lw. ins C-Register
            call   ReadDC         ; DiskChange einlesen
            and    0000001b       ; Bit 1 abfragen
            ld      de,MsgAlt
            jr     nz,PrtStr
            call   ResetDC        ; DiskChange zurücksetzen
            ld      de,MsgNeu
PrtStr:     ld      c,9
            call   5              ; Ausgeben und Heim
            ret

MsgAlt:     db      'Diskette wurde nicht gewechselt$'
MsgNeu:     db      'Diskette wurde gewechselt$'

false equ 0
true  equ not false
fdx   equ true          ; FDX

        if     fdx
FdcPort equ 40h         ; fd-controller port
        else
FdcPort equ 10h        ; sdx-controller port
        endif

ReadDC   ; lies DiskChange-Signal ein
        ld    a,c
        if   fdx          ; selektiere Laufwerk
            and    3
            or     00001100b ; Dma disable, 8", DD, DrvSel
            out   (FdcPort+4),a
            out   (FdcPort+5),a ; DriveSelect-Change
        else
            and    1          ; nur zwei Drives
            or     00011100b ; motor on u.s.w.
            out   (FdcPort+4),a
        endif
        in     a,(FdcPort+5) ; DC-Signal
        push  af            ; merke Wert

        ld    a,c          ; drive sel. disable
        out   (FdcPort+4),a
        pop   af
        ret

```

H a r d w a r e: Abfrage, ob Diskette gewechselt wurde

```

ResetDC:          ; DiskChange-Signal zurücksetzen

                  ; erst wird ein DriveSelect über das BIOS gemacht, und
                  ; dabei dem BIOS mitgeteilt, daß es sich nicht um das
                  ; Wieder-Anwählen, sondern um ein Neu-Anwählen handelt
                  ; Wie man dsa BIOS aufruft: siehe weiter oben in diesem Info
push             bc                ; Register retten
ld              hl,(1)             ; Adresse WarmBoot
ld              de,18h
add             hl,de              ; Adresse DriveSelect
ld              e,0                ; WICHTIG! Neu-Anwählen
inc             c                  ; da BIOS A: = 0 will
call           CallHL
pop             bc

                  ; nun der STEP
ld              a,c
if              fdx                ; selektiere Laufwerk
and             3
or              00001100b         ; Dma disable, 8", DD, DrvSel
out             (FdcPort+4),a
out             (FdcPort+5),a    ; DriveSelect-Change
else
and             1                  ; nur zwei Drives (hihi!)
or              00011100b         ; motor on u.s.w.
out             (FdcPort+4),a
endif
ld              a,01011000b       ; Step-In
out             (FdcPort+0),a
call           waitb              ; warte bis das geschafft

ld              a,c                ; drive sel. disable
out             (FdcPort+4),a
ret

CallHL:          jp              (hl)

pause:
if              fdx
ld              a,255              ; 32h
else
ld              a,32h
endif

paus01:         dec             a
jp              nz,paus01
ret

waitb:          call           pause ; wait while busy
in              a,(FdcPort+0)
and            00000001b          ; 01h
jr              nz,waitb
ret

end

```