

# CLUBINFO

36. Ausgabe

Kontaktadresse: Club80 / Fritz Chwolka / Saarstraße 34 / 5173 Aldenhoven  
Tel.: 0 24 64 / 89 20

# Inhaltsverzeichnis

## Clubinternes

	<u>Autor &amp; Seite</u>
Vorwort	1
"Formatknackerbande"	Fritz Chwolka 2
Termine	Uwe Schobert 2
	Redaktion

## Software

Noch serieller mit PIP	3 - 4
Prozessor piep einmal ...	5
BONUM BENE	Alexander Schmid 6 - 15
Ein bißchen C, ein bißchen Mathe...	Volker Dose 17 - 23
	Kurt Müller

## Hardware

Umrüstung ade !	24
CPU280 - die Z80-VAX	Uwe Schobert 25 - 26
Einfachste A/D- und D/A-Wandler	27 - 28
	Alexander Schmid

## Börse

Wer hat was – wer will was	29 - 30
----------------------------	---------

## Sonstiges

Der Rechner mit der Handbremse	31 - 33
Schönere Schildchen	33
Hacker's Hirnkrücke	34
	Alexander Schmid
Exceptions, Interrupts und Traps	35 - 38
Vom Single zum Multi	39 - 45
	Gerald Schröder
XT - Extra viel Trouble ???	47 - 48
	Kurt Müller
Keine Glosse, sondern Wirklichkeit	49 - 50
	Klaus-Jürgen Mühlenbein
Public Domain Software	51 - 54
	Fritz Chwolka

## Die letzten Seiten

Impressum	55
Schluß	56
	Redaktion
Mitgliederadressenliste	am INFO-Ende

Hallo Freunde!

Diesmal möchte ich nicht über fehlende Manuscripte schreiben, auch wenn der Jens sich bitter darüber beklagt. Aber warum bringen denn so wenige von EUCH ihre Gedanken zu Papier? Es muß doch nicht immer ein super-hyper-megabytemäßiges Selbstbauprojekt sein!

Aber ich wollte ja etwas anderes ansprechen. Der Club nennt sich CLUB 80, weil er sich dem Z80 und damit verbundener Peripherie widmet. Nun hat aber fast jeder von uns neben einem Z80 Rechner auch Kontakt zu z.B. einem MSDOS-kompatiblen Computer, einem MAC oder auch einem Atari. Ich möchte auch nicht verschweigen, daß ich neben einigen CP/M-Rechnern auch einen AT habe.

Meiner Meinung nach können wir uns durchaus in Artikeln oder Fragen auf diese Rechner beziehen, wie es auch im letzten oder vorletzten Info geschehen ist. Durch unsere Arbeit mit dem Z80, ich meine natürlich unser Hobby, haben wir mehr von Rechnern und deren Struktur kennengelernt als es einem normalen Anwender je gestattet ist. Warum sollen wir denn nicht auch etwas darüber mitteilen.

Hierdurch wird der Grundgedanke des Clubs nicht umgestoßen, sondern vielleicht der Rahmen etwas weiter gesteckt.

Auch dürften die Artikel hierzu nicht überhand nehmen, da doch an jedem Kiosk die Computerhefte kiloweise erhältlich sind und Ihr hoffentlich nicht abschreibt.

#### Hier nun ein Gedanke und Aufruf zur Hauptversammlung

Einer der Gedanken des Clubs ist, mindestens einmal im Jahr eine Hauptversammlung abzuhalten. Mein terminlicher Vorschlag wäre der Monat März oder April. Da die Familie Obermann aus beruflichen Gründen diesmal wahrscheinlich als Veranstalter ausfällt (schade), sollte sich einer von EUCH dazu bereiterklären. Die Aufgabe ist ja klar, durch Rücksprache mit Vorstand den Termin abklären und einen geeigneten Ort mit Unterkunftsmöglichkeit suchen. Auch sollte der Veranstaltungsort so ziemlich in der Mitte unseres Einzugsgebietes liegen. Ich hoffe auf reichliche Anrufe und habe schon ein extralanges Band in den Anrufbeantworter gesteckt.

*Fritz Chwalke*  
Fritz Chwalke

Liebe Clubfreunde!

Nachdem Hans-Günther Hartmann in seinem Bericht '... Formatknackerbande ...' den einzigen erwähnt hat, der einen AlphaTronic PS betreibt und den er kennt, so möchte ich mich gleich freiwillig melden und ihm für diesen guten Artikel danken.

Dies ist zumindest ein erster Schritt für mich, andere Diskettenformate zu lesen, da mein Betriebssystem das leider gar nicht unterstützt. Ich weiß, daß da noch viel auf mich wartet, doch vielleicht löse ich auch das einmal. (Es ist erstaunlich, wie sehr einen eine solche Schwierigkeit reizt, wo manch anderer Computer das doch im Handumdrehen schon kann.)

Ich hoffe jedenfalls, nicht zuletzt mit Eurer indirekten oder direkten Hilfe, irgendwann einmal unser Club-Diskettenformat (gibts denn ein einheitliches?) direkt lesen zu können.

Mit freundlichen Grüßen

Uwe Schoberth

Tel.: priv. 07041/7254

*Uwe Schoberth*

#### Termine... Termine... Termine... Termine... Termine

Jahreshauptversammlung 92	.....	Mai...- .....92
Clubtreffen Süd	.....	.....- .....92
DOMOTECHNICA	Köln	18.02.- 21.02.92
CeBIT	Hannover	11.03.- 18.03.92
CAT-Anwenderkongreß	Stuttgart	05.05.- 08.05.92
Orgatec	Köln	22.10.- 27.10.92
Electronica	München	10.11.- 14.11.92

Redaktionsschluß für die Clubinfo's ist jeweils der letzte Tag der folgenden Monate:

März 92  
Mai 92  
Juli 92

Club 80  
INFO 36  
Jan. 92

Seite  
2

Club 80  
INFO 36  
Jan. 92

Seite  
1

## Noch serieller mit PIP

Im Info Nr. 35 hat "uns Fritz" geschrieben, wie man im Notfall auch ohne kompatible Laufwerke, bzw. wenn keine Informationen über das Diskettenformat vorliegen, Dateien mit Hilfe von PIP von einem Rechner zum anderen übertragen kann. Dazu noch zwei kleine Anmerkungen/Ergänzungen:

Wenn man PIP als Parameter für die Übertragung zusätzlich noch ein "H" (Hex) mitgibt, schaltet es in den Hex-Modus und überprüft, ob mit der übertragenen Datei (in der Z80-Welt üblicherweise im sog. Intel-Hex-Format) alles stimmt. Diesen Service hat man nur mit "B" (Block) alleine nicht und kann verfälschte Zeichen höchstens daran erkennen, daß der HEX-COM-Wandler (HEXCOM, LOAD o.ä.) hinterher meckert, oder das Programm nicht läuft. Im Hex-Modus stellt PIP anhand der Prüfsumme in jeder Zeile fest, ob möglicherweise ein Fehler aufgetreten ist und meldet dies dann sofort. Wenn man nur beim sendenden Rechner ein E (Echo) als Parameter angibt, wird dieser weiter gebremst, was dem empfangenden Rechner etwas mehr Zeit gibt, die ankommenden Zeichen zu verarbeiten. Normalerweise dürfte das nur bei einem relativ langsamen Rechner auf der einen und einem schnelleren Rechner auf der anderen Seite notwendig sein, aber aus Erfahrung würde ich sagen, daß man das immer so machen sollte. Auch ist es nicht unbedingt gesagt, daß es mit einer kleineren Baudrate immer besser geht, ich habe es schon erlebt, daß es mit 1200 Baud einwandfrei ging, mit 300 Baud aber nur Schrott angekommen ist. Warum? Keine Ahnung, aber wenn alle Geheimnisse gelöst wären, wäre es ja langweilig. Außerdem sollte man beim sendenden Rechner die Schnittstelle tunlichst auf das XON/XOFF-Protokoll einstellen, sonst funktioniert der Blockmodus nicht. PIP sendet zwar XOFF- und XON-Zeichen, aber die Übertragung wirklich anhalten muß der Schnittstellentreiber bzw. das BIOS. Wenn sich der empfangende Rechner nach der Übertragung nicht mehr zurückmeldet, kann man ihm mit "PIP PUN:=EOF:" bzw. "PIP OUT:=EOF:" noch ein explizites Dateiende zukommen lassen.

Was in Fritz' Artikel vielleicht auch nicht ganz klar rausgekommen ist und viel Zeit und Nerven kosten kann: Der Test mit

PIP PUN:=filename[EB] und PIP filename=RDR:[EB]

(hinter dem Filenamem darf, entgegen der Angabe von Fritz, übrigens KEIN Doppelpunkt kommen, sonst gibt's eine Fehlermeldung !!)

funktioniert nur, wenn das CP/M das IO-Byte unterstützt! Es gibt Implementationen, die das Byte einfach nicht benutzen und dann wundert man sich, daß nichts geht. Ausprobieren kann man das mit "PIP LST:=CON:", mit dem alle Eingaben von der Tastatur auf den Drucker umgelenkt werden sollten. Da LST, genauso wie EOF weiter oben, ein logisches Gerät ist, muß hier zur Unterscheidung von einem Filenamem nun tatsächlich ein Doppelpunkt dahinter stehen. Wenn man aber ein komfortables BIOS hat, das u.a. bei einem leeren Empfangspuffer brav selber auf ein neues Zeichen wartet und der Testtext so fehlerfrei angekommen ist, ist man schon fertig und braucht die ganze Patcherei garnicht. Dann kann man mit PUN: bzw. RDR: arbeiten und im Prinzip sofort loslegen und beim CP/M Plus, wo man in beiden Fällen nur über AUX: geht, muß man sowieso nicht fummeln und kann auch gleich anfangen.

Wenn es aber doch sein muß, sei es weil das mit dem IO-Byte nicht funktioniert, oder dauernd Zeichen ankommen, ohne daß der andere Rechner

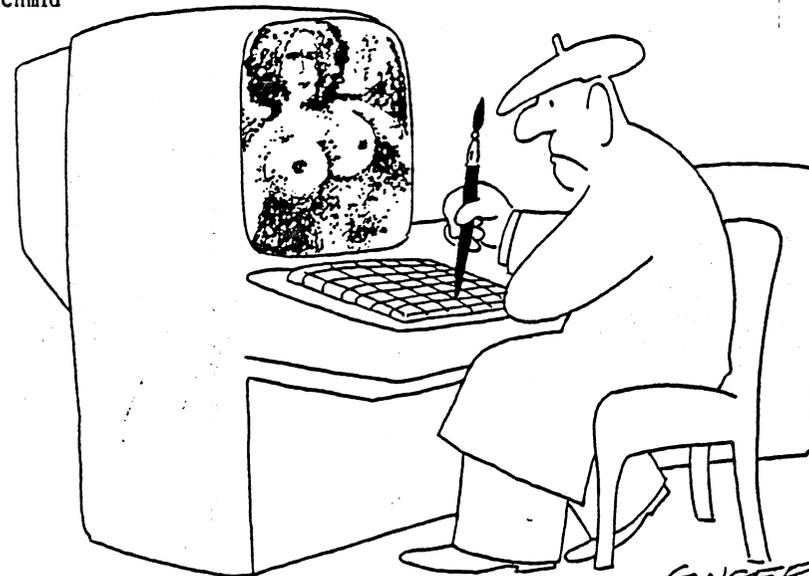
sender, und man PIP schließlich so gepatcht hat, daß es die Ports direkt anspricht, muß die Ein- und Ausgabe folgendermaßen aussehen: (PUN: und RDR: gehen nach wie vor über das Betriebssystem, da helfen alle Patches nichts !)

PIP OUT:=filename[EBH] und PIP filename=INP:[BH]

Wenn diese Hürde dann schließlich genommen ist und die Übertragung klappt, sollte man so schnell wie möglich LYNC überziehen. Das Programm besteht aus dem eigentlichen COM-File (LYNC.COM), einem Installer (LYNC-INST.COM) und einer Treiber-Bibliothek (IO.DRV). Am besten konfiguriert man das Programm schon vorher, dann braucht man nur ein File zu übertragen. Das File ist auch als Hex-File so klein, daß es zusammen mit PIP in den Arbeitsspeicher paßt, sodaß im Block-Modus erst nach Ende der Übertragung auf die (langsame) Floppy zugegriffen werden muß und es so keine Probleme mit dem Protokoll geben kann.

Warum aber nun ausgerechnet LYNC? Ganz einfach, es ist verhältnismäßig klein und so ziemlich das einfachste und sicherste Programm für diesen Job, bei dem man eigentlich nichts falsch machen kann. Der Installer arbeitet menügeführt und man muß nur die Ports der seriellen Schnittstelle, die man ja eventuell schon von den Patches in PIP kennt, per Hand eingeben; für viele Rechner (u.a. TRS-80 Model I und IV) gibt es einen fertigen Menüpunkt, den man nur anzuwählen braucht. Danach kann man von einem der beiden Rechnern beliebig Files senden und empfangen, ohne daß am anderen Rechner jemand sitzen muß. Man kann sich z.B. sogar das Directory des anderen Rechners ferngesteuert ansehen und dann mit SEND und FETCH loslegen. Das ist viel bequemer als mit Kermit oder sonstigen Terminalprogrammen, wo man immer noch einen Helfer braucht. Die Übertragung wird durch ein LYNC-eigenes Protokoll gesichert und nichtmal das mutwillige Unterbrechen der Leitung bringt es aus dem Tritt. Nachdem die Verbindung wieder steht, geht es sofort weiter. Außerdem überprüft sich LYNC beim Starten selber, sodaß man dadurch nochmal eine Kontrolle hat, ob die Übertragung des Hex-Files fehlerfrei war.

Alexander Schmid



SANGER

## Prozessor piep einmal...

wenn ein Programm auf einen bestimmten Prozessor angewiesen ist, ist es ganz nett, wenn man auf dem Bildschirm wenigstens eine Meldung ausgibt, daß das Programm z.B. einen Z80 braucht und auf einem 8080 nicht lauffähig ist, statt kommentarlos ins Betriebssystem zurückzuspringen. Für den Z80 und HD64180 gibt es schon einige Lösungen, aber für den Z280 ist mir noch keine über den Weg gelaufen. Wenn man ein neues Feature des Z280 (die Queue) ausnutzt, ist das ohne großen Aufwand mit folgender Routine möglich:

```
XOR A
LD (flag),A
LD A,1
flag EQU $-1
```

Auf einem "normalen" Z80 wird der Akku in der ersten Zeile gelöscht, in der zweiten Zeile wird diese Null in den Ladebefehl in Zeile drei gepatcht und in Zeile drei bekommt der Akku folglich den Wert 0. Soweit ganz klar und verständlich, aber nicht so beim Z280, wenigstens auf den ersten Blick. Der Z280 hat nämlich die Eigenschaft, daß er immer etwas "vorausschaut", d.h. schon die nächsten Bytes lädt, wenn er den aktuellen Befehl gerade ausführt. Im Beispiel heißt das, daß er, wenn er Zeile zwei ausführt, schon den Code von Zeile drei in die Queue (Warteschlange) geladen hat und daher gar nicht mehr mitbekommt, daß da inzwischen ein anderer Wert hineingeschrieben wurde. Sowa ist natürlich nicht gerade die feine englische Art, aber im Normalfall macht man das ja auch nicht, oder? Bei einem Z80 enthält der Akku also anschließend eine 0, bei einem Z280 eine 1 (wie das dann weiter verarbeitet wird ist hier nebensächlich). Ganz nebenbei bemerkt ist das auch ein prima Debug-Schutz, der Trick mit der Queue funktioniert nämlich nur, wenn das Programmstück auf einmal durchlaufen wird. Wenn ein Debugger im Einzelschritt-Modus hinter jedem Befehl einen Sprung einfügt wird die Queue jedesmal geleert und der Prozessor verhält sich im obigen Beispiel wie ein gewöhnlicher Z80. Bei einem speziellen Z280-Programm würde man dann immer mit einer Fehlermeldung im DOS landen und sich fürchterbar wundern, warum der Z280 einfach nicht erkannt wird. Im Prinzip gilt das auch für den 80x86, nur variiert dort die Länge der Queue von Typ zu Typ.

## BONUM BENE

Obwohl die Benutzung von G-DOS/NewDOS80 oder auch Calvados eine Kategorisierung der User in die Sparte "Ewissestrise" durchaus gerechtfertigt, hat dieses Betriebssystem auch seine Vorzüge.

Die ganzen schoenen Features die moderne Betriebssysteme aufweisen, fehlen bei uns zwar, aber man kann sie wenigstemns noch selber programmieren. Zumindest einige.

Mich hat beim CP/M und MS-DOS immer die Benutzung von Wildcards, also \* und ?, bei Ansabe von Filenamen beeindruckt.

Insbesondere beim Kopieren von grossen Datenmengen zeigen sich im Copy-Befehl des NewDOS ernsthafte Schwächen. Zum einen werden vor dem Kopiervorgang die Filenamen schon ins Directory eingetragen: wenn die Diskette dann aber voll ist, sind die Files 'leer', man muss dann von Hand die nicht kopierten Files suchen und dann File fuer File loeschen. Und die Einsabemoeslichkeit bei dem Parametern ',EDK,FRAG,...' ueberzeugt mich auch nicht immer. Andauernd sehe ich ein 'J' oder auch mal ein 'N' zuviel und ich muss mir den File dann aufschreiben, um ihn hinterher zu loeschen oder zu kopieren. Je nach dem. Und besonders praktisch ist das Sternchen, wenn man viele Files mit dem selben Anfangsbuchstaben hat die in einem ursaechliche Zusammenhang stehen, bei mir zum Beispiel Apfelmaenchendateien, die APFEL/CMD, und dann eben APFEL1/GRF bis APFEL29/GRF und dann noch die /DAT oder /NEW oder /MHM Dateien in welchen andere Daten oder die Bilder selbst verstecken. Oder ein anderes Beispiel, Kopiere alle Dateien die Sourcecodes von Sysfiles sind: hiesse dann Kopiere SVS\*/SRC.

Nun ja, meine ersten Versuche machte ich in Basic. Aber trotz Compilierungs war es einfach unertraeslich langsam, es nutzte also nichts, Assembler musste gelernt werden. Sehr geholfen haben mir dabei das DOS-Buch von H. Grosser sowie die kommentierten Assembler-routinen die sich immer wieder in alten Club 80-Infen finden.

Also fluss ZEUS angeworfen und los gines. Sehr schnell stellte ich fest, das sich Z80-Assembler von Hochsprachen doch ziemlich unterscheidet, besonders die Beherrschung der bedinsten Spruense und Unterprogrammaufrufe machte mir zunaechst ziemliche Schwierigkeiten (ist das Z-Flas nun gesetzt oder nicht oder was?). Aber nach einisen Tagen hatte ich dann endlich ein lauffaehises, narrensicheres und vor allen Dingen sauschnelles Kopierutility vor mir.

Nun zum Programmablauf noch einise Erlaeuterungen. Die ersten 19 Zeichen des Programms sind reserviert und werden vom Programm mit Parametern sefuellt. Die Einsabe wird in ein Format uebertragen das dem Format des FDE (File Directory Eintrag) im Inhaltsverzeichnis der Disketten gleicht, also immer 11 Zeichen und kein Schraesstrich. Eventuelle Sterne im

Einsabestrings werden in Frasezeichen umgewandelt. Danach werden die Laufwerksnummern von Source- und Destinationdrive aufgefunden und eingetragen. Wenn die Einsabe fehlerhaft war, wird eine Fehlermeldung mit der richtigen Syntax ausgeben.

Die Anzahl der FDE-Sektoren einer Diskette, in der maximal 8 Eintraese fuer Files stehen koennen, findet sich im ersten Sektor des Inhaltsverzeichnisses in Byte 1Fh. Die Zahl wird ermittelt und eingetragen. Es werden genau soviele FDE-Sektoren mit dem Dosunterprogramm 'DIRR' geladen und wenn der Eintraes besteht (gesetztes bit 4 des ersten Bytes des Eintraeses) und es sich weder um ein System-File (gesetztes bit 6) noch um einen Folseeintraes (gesetztes bit 7) handelt, wird der Filename mit den 11 Bytes am Anfang des Programmes, in dem der geforderte Filename steht, verglichen. Dabei silt ein Frasezeichen soviel wie jedes anders Zeichen, welches als Filenamezeichen erlaubt ist (Ziffern und Umlaute sowie beim Genie IIIs der untenstehende Strich).

Entspricht der Filename im Directory den geforderten Spezifikationen, dann wird er in einen Buffer am Ende des Programmes eingetragen. Davor wird aber Jedesmal ein '>', dies ist der COPY-Befehl des G-DOS eingetragen. Hinter den Filename wird noch 'LW(source),LW(dest)<ENTER>' eingetragen, um den Befehl mit einem Dos-Call ablaufen zu lassen. Dieser Vorgang wird wiederholt, bis alle FDE-Sektoren der Source-Diskette gelesen sind, so das sich am Programmende, in dem Buffer, eine Reihe von zueltsigen Doskommandos (kann man sich mit DEBUG sehr schoen anschauen) befindet. Der Anfang des Buffers ist in 'buf2' zwischenspeicherter, 'buf2' zeigt also auf den ersten Kopierbefehl. Sofern die Anzahl der zu kopierenden Files nicht null ist (ersibt dann eine Meldung) oder die Einsabe 'Kopiere \*.\* ...' gesetzt wurde (bewirkt eine nochmalise Nachfrage 'Sollen alle Files kopiert werden', die mit einem grossen 'J' beantwortete werden muss) wird ein Kopierkommando nach dem anderen zunaechst angezeigt und dann ausgefuehrt.

Sollte eine Fehlermeldung auftreten, die eine volle Diskette oder ein volles Inhaltsverzeichnis anzeigt, dann wird der letzte zu kopierende File wieder geloescht, da er auf der Zieldiskette nicht ordnungssmaess abgespeichert wurde.

Der einzise Fehler, den das Programm aufweist ist das im Einsabestring keine Ziffern fuer die Filespezifikation benutzt werden koennen. Das duerfte aber auch relativ nebensaechlich sein.

Wenn man das Programm anders nennen moechte, muss man im Sourcelistings in Zeile 330 eine Aenderung vornehmen. Der Eingabepuffer des DOS liest ab Speicherstelle 4318h. Hier steht nachdem man <ENTER> eingeseben hat die vor dem ENTER liegende Einsabe, in diesem Fall zum Beispiel 'KOPIERE \*.\*:0,:1'. Die Erkennung der Einsabe '\*.\*' sowie die Behandlung derselben habe ich folgendermassen seloest: An der Speicherstelle 4320h steht der erste der Sternchen. Wenn das Programm umbenannt ist, zum Beispiel in 'KOPY' wuerde ab 4318h folgendes stehen: 'KOPY \*.\*:0,:1'. In Speicherstelle

4320h steht dann der Doppelpunkt vor der Null, das Programm kann den String '\*.\*' also gar nicht erkennen. Die Zahl in 330 muesste in diesem Fall also 4310h heissen. An dieser Stelle befindet sich dann der erste Stern. Sinnemaess silt das auch fuer die Zeilen 333 und 337. Hier muessen die Zahlenwerte jeweils incrementiert werden

Das Grundueruest dieses Programmes liesse sich fuer viele andere Geschichten nutzen, die Implementation eines 'ERASE B\*.\*' um Filegruppen zu loeschen (zum Beispiel genau die, welche gerade auf eine Sicherheitskopie mit 'KOPIERE' kopiert wurden) oder auch 'INU' bzw 'UIS' um Files unsichtbar oder sichtbar zu machen oder auch nur um unsichtbare Files anzuzeigen (und zwar nur die) waeren meiner Meinung nach huebsche Features. Wahrscheinlich fallen Jedem von euch auch noch andere Geschichten ein, je nach dem was euch gerade beschaeftigt.

Das Programm ist so fuer Jeden Rechner nutzbar, auf dem irgendeine Version des G-DOS laeuft, weil ich nur ziemlich allgemein gehaltene DOS-Routinen aus dem SYS0/SYS benutze. Das sollte auf Jedem G-DOS gleich sein.

Die NewDOS80-Benutzer muessen die Routine 'ein>' ab Zeile 274 bis Zeile 280 loeschen und die ':' also die Kommentarzeichen vor den Zeilen 285 bis 300 loeschen, so das dieser Teil jetzt die Routine 'ein>' darstellt.

Wer das Programm nicht abtippen will, kann mir eine Diskette im G-DOS Standardformat fuer 80-Spuren DS/DO schicken, und denn auch bitte Rueckporto beifuehen.

Noch mal die PDRIUE-Parameter:  
TI=EHK TD=8 SP=80 SEK=36 SWZ=0 EIB=6 SBIU=48 AEIU=2  
Tja, in diesem Sinne  
'DEM GUTEN DAS BESTE'  
Happy Hackins !!!!!

PS.: In einer lauschisen Stunde habe ich das Sourcelistings umgeschrieben, auf das es jetzt die Funktion 'ERASE' darstellt. Die Hauptarbeit las darin, die Kommentarzeilen umzuschreiben, der Rest ist wirklich ziemlich einfach.

Statt der Routine 'ein>' muss eine Routine beschrieben werden, welche den Befehl 'KILL' in den Buffer einschreibt. Ausserdem muessen die Stellen, wo Source- und Destinationlaufwerksnummern eingeschrieben oder abgefragt werden selendert werden. Das Programm 'ERASE' werde ich auch zum Diskothekar schicken.

PPS.: Seit einigen Wochen bin ich im Besitz eines GENIE IIIs mit dem 1MByte-Umbau von Helmut Bernhardt. Sehr schade finde ich, dass es keinen vernuenftigen Grafikcompiler bzw. eine Library fuer den BASCOM sibt. Ich habe zum Beispiel fuer das Genie I im HRG-Pack-Paket eine Library fuer BASCOM. Jetzt hatte ich die Idee, diese File zu disassemblieren und umzuschreiben auf das es die HRG des G IIIs beharken moese. Wenn Jemand schon mal an soetwas gedacht hat oder es gar

beschafft hat oder es versuchen will oder auch nur so, kann er oder sie sich ja mal bei mir melden.  
PPPS.: Ich suche noch eine BASCOM-Version fuer Genie-DOS, welche CMD-Files generiert, mein Linker erzeugt nur CHN-Files die mit BRUN gestartet werden.

```

00001 ; K O P I E R E
00002 ;
00003 ;Kopieren von einer diskette auf eine andere
00004 ;mit ansabe von wildcards
00005 ; ? = beliebiger buchstabe
00006 ; * = nachfolgende zeichen bis '/' beliebis
00007 ;
00008 ;
00009 ; Syntax := kopiere FILENAME/EXT:lw#:lw#<ENTER>
00010 ;
00011 ;
00012 ;
00013 outch EQU 0033h
00014 setkey EQU 0049h
00015 dosrdr EQU 402dh
00016 doscal EQU 4419h
00017 extrfs EQU 441ch
00018 sendms EQU 4467h
00019 srosbu EQU 45b5h
00020 dirr EQU 490ah
00021 dirw EQU 491fh
00022 tesdsk EQU 445eh
00023 debus EQU 440dh
00024 doserr EQU 4409h
00025 ;
00026 ;
00027 ORG 8000h
00028 dri1 DM 'x' ;source-lw#
00029 dri2 DM 'x' ;destination-lw#
00030 secanz DM 'x' ;anzahl der zu lesenden
;dirsectoren mit FDE's
00031
00032 buf1 DM 'xx' ;zeiser auf buffer
00033 buf2 DM 'xx' ;zeist auf anfang Bufferfeld
00034 filzae DM 00h ;Anzahl der zu kopierenden Files
00035 fname DM 'FILENAME';ebenso vom Prosr. eingesetzt
00036 exten DM 'EXT'
00037 ende DM 0DH ;in den zurueckklingenden 11 bytes
;wird der Vergleichsfilespec
;eingeschrieben um verslichen
;zu werden
00041 schlech DM 'Syntax := KOPIERE FILENAME/EXT'
00042 DM ':<lw#s>,<lw#d>ENTER'
00043 DM 0dh ;Bei falscher Einsabe
00044 start LD B,09h ;max. 8Zeichen f>r Filename
00045 LD DE,fname; Zeiser auf
00046 loop1 LD A,(HL) ; zeichen holen
00047 CALL srosbu ; a->A, soweit moelich
00048 CP '?' ; wenn ja einfach eintragen
00049 JR Z,weil
00050 CP 'A'
00051 JR C,kzeich; wenn Carry gesetzt, ist
;das Zeichen in der
;ASCII-Tabelle kleiner als 'A'
00052
00053
00054 CP '['
00055 JR NC,kzeich; bzw. es ist oberhalb von 'Z'
00056 weil LD (DE),A ; buchstaben eintragen
00057 INC DE
00058 INC HL ; zeiser einen weiter
00059 DJNZ loop1 ; bis 8 Zeichen
00060 error ; mehr als 8 !!
00061 kzeich CP ':' ; laufwerkstrennzeichen?
00062 JR Z,error ; enter ohne ext ist unzulässig
00063 CP '/' ; einziges zulässiges zeichen
00064 JR Z,slash ; weiter im text
00065 CP '*' ; Jetzt wirds interessant
00066

```

```

00067 JR error ; anderes zeichen !!
00068 slash LD A,B ; wegen DJNZ
00069 CP 0h ; 8tes Zeichen ?
00070 CALL NZ,nulauf; ausnullen mit spaces
00071 LD DE,exten ; zeiser auf extensionstring
00072 ext LD B,04h ; hichstens drei Buchstaben !
00073 loop2 INC HL ; eiansns z(hler ein weiter
00074 LD A,(HL)
00075 CALL srosbu ; selbe routine wie oben
00076 CP '?' ;einfach weiter
00077 JR Z,weil2
00078 CP 'A'
00079 JR C,keexze; wenn enter, dann o.k.
00080 CP '['
00081 JR NC,keexze; dito
00082 weil LD (DE),A ; buchstabe eintragen
00083 INC DE
00084 DJNZ loop2 ; bis 3 Zeichen
00085 CP ':' ; 13. Zeichen-LW-Doppelpunkt ?
00086 JR NZ,error; wenn nicht !!!!!!!!!!!!!
00087 JR kopie ; soweit in ordnung
00088 keexze CP '*' ; mit ?? aufüllen
00089 JR Z,sternd
00090 CP ':'
00091 CALL NZ,error; Jetzt folst die LW#
00092 CALL nulauf
00093 JR kopie
00094 nulauf LD A,' ' ; space
00095 fill LD (DE),A ; in fname eintragen
00096 INC DE
00097 DJNZ fill
00098 RET
00099 stern LD A,?' ; fuellbyte '?'
00100 CALL fill
00101 INC HL ; m)~te auf '/' zeisen
00102 JR slash
00103 stern1 LD A,?' ; f)llbyte
00104 CALL fill ; ausfüllen,
00105 INC HL ; soll auf ':' zeisen
00106 LD DE,ende ; aufs ende von fname
00107 LD A,(HL) ; zeichen holen
00108 CP ':' ; wegen LW#
00109 JR NZ,error
00110 JR kopie ; soll funktionieren
00111 error LD HL,schlech
00112 CALL sendms ; error meldung ausgeben
00113 JP dosrdr ; vorlaufiser aussties
00114 istzahl SUB '0' ; ascii korrektur
00115 JR C,error
00116 CP 10
00117 RET C
00118 JR error
00119 kopie INC HL ; Jetzt wird erst mal auf
; die Befehlssyntax
00120
00121 LD A,(HL) ; also :lw#:lw# getestet
00122 CALL istzahl
00123 LD (dri1),A; als source laufwerk
00124 INC HL ; auf komma
00125 LD A,(HL)
00126 CP ','
00127 JR NZ,error; keine andere einsabe erlaubt
00128 INC HL ; auf zweiten doppelpunkt
00129 LD A,(HL)
00130 CP ':'
00131 JR NZ,error
00132 INC HL ; auf destination laufwerk

```

```

00133 LD A,(HL)
00134 CALL istzahl
00135 LD (dri2),A; als dest-lw
00136 INC HL ; auf ENTER
00137 LD A,(HL)
00138 CP 0dh
00139 JR NZ,error;
00140 ; weiter im text, die Eingabe ist richtig
00141 ; und der vergleiche filename steht.
00142 ; die laufwerksnummern sind auch da,
00143 ; Jetzt kann wirklich kopiert werden.
00144 ; es werden ersteinmal die eintr(ge in den
00145 ; buffer beschrieben
00146 LD HL,buffer
00147 LD (buf1),HL
00148 LD (buf2),HL; anfang des Buffers
00149 ; muss erhalten bleiben
00150 reasec LD A,(dri1); sourcelaufwerk eintragen
00151 CALL tesdsk ; disketten zugriff testen
00152 JP NZ,doserr
00153 CALL ermi ; wieviele Dir-sectoren
00154 LD A,(secanz)
00155 LD B,A
00156 LD A,1 ; erster dir-sector mit fde's
00157 liesec INC A ; ist der Sector Nummer 2
00158 PUSH AF ; retten !!
00159 PUSH BC ; abenso
00160 CALL dirr ; les einen sector in DOS-buffer
00161 CP 06h ; 'Lesevers. markierter satz'
00162 JR Z,wei3 ; dann ist alles in ordnung
00163 JP doserr ; es ist ein fehler aufsetr.
00164 wei3 CALL auswert; seltise filespecs errach.
00165 POP BC
00166 POP AF ; accu vom stack als zähler
00167 DJNZ liesec
00168 JP copy ; Jetzt kann es richtig lossehen
00169 auswert LD B,7 ; acht eintr(ge pro sector
00170 LD HL,4200h; erstes byte zeigt ob
00171 ; ein Eintrag vorhanden ist
00172 LD DE,4205h; zeigt auf ersten FILENAMEEXT
00173 CALL verslei ; prüft ob der eintrag den
00174 loop3 LD A,20h ; spezifikationen
00175 ADD A,L ; entspricht. Ein FDE ist
00176 LD L,A ; 20h Byte lang.
00177 LD A,20h
00178 ADD A,E
00179 LD E,A
00180 CALL verslei ; siehe 8 Zeilen weiter oben
00181 DJNZ loop3
00182 RET ; acht eintr(ge sind serr)ft
00183 verslei PUSH HL ; HL und DE werden gerettet
00184 PUSH DE
00185 PUSH BC ; mu* auch gerettet werden
00186 LD B,0bh ; 11 zeichen ist fname lang
00187 LD A,(HL) ; sehen ob eintrag besteht
00188 CP 00h ; null, wenn nicht
00189 JR Z,zur ; erst noch pop de und hl
00190 CP 90h ; ist es ein Folgeeintrag ???
00191 JR Z,zur ; dann nicht eintragen !!
00192 BIT 6,A ; bit 6 ist '1' bei /SYS-Files
00193 JR NZ,zur ; weil die nicht kopiert
00194 ; werden dürfen (oder sollen)
00195 ; vor allen Dingen nicht Boot und
00196 ; Inhalt/SYS !!!!!!!!!!!!!!!!!!!!!!!
00197 LD HL,fnam; enth(lt spezifizierten fname
00198 wei4 LD A,(DE) ; zeichen im buffer

```

```

00199 CP (HL) ; vergleiche (hl) mit (de)
00200 JR NZ,wei7 ; wenn nicht gleich, prüfen ob ?
00201 wei5 INC HL
00202 INC DE ; zeiger auf nächstes zeichen
00203 DJNZ wei4 ; 11 zeichen vergleichen
00204 eintr NOP ; alle zeichen entsprechen den
00205 ; spezifikationen, jetzt müssen
00206 ; sie in einen Buffer
00207 ; eingetragen werden, allerdings
00208 ; in der Form FILENAME/EXT
00209 LD A,E ; DE mu* um 11 erniedrigt werden
00210 SUB 11 ; damit DE wieder auf Anfang des
00211 LD E,A ; filespecs im FDE zeigt
00212 LD A,(filzae); zähler für zu kop. Files
00213 INC A ; + 1
00214 LD (filzae),A ; und speichern
00215 LD HL,(buf1); buf1 ist der zeiger auf buffer
00216 CALL ein ; der DOS-Befehl '>' wird
00217 ; vor die filespecs
00218 ; eingetragen wesen doscall
00219 LD B,8 ; acht zeichen höchstens
00220 loop4 LD A,(DE) ; erstes zeichen im buffer
00221 CP ' ' ; ist es space ?
00222 JR Z,nein
00223 LD (HL),A ; zeichen eintragen
00224 INC HL
00225 nein INC DE ; zeiger weiter
00226 DJNZ loop4 ; bis 8 zeichen
00227 LD A,'/'
00228 LD (HL),A ; Trennzeichen eintragen
00229 INC HL
00230 ext2 LD B,3 ; 3 extension zeichen
00231 loop5 LD A,(DE) ; erstes zeichen
00232 CP ' '
00233 JR Z,nein2
00234 LD (HL),A
00235 INC HL
00236 nein2 INC DE
00237 DJNZ loop5
00238 DEC HL ; abfrage, ob gar keine extension
00239 LD A,(HL) ; vorhanden war, dann mu* der '/'
00240 CP ' ' ; nämlich auch wieder was
00241 JR Z,wei6
00242 INC HL
00243 wei6 LD A,';' ; die zeichen ':lw#,:lw#',0dh
00244 LD (HL),A ; werden hinter jeden filename
00245 INC HL ; eingetragen
00246 LD A,(dri1)
00247 ADD A,48 ; ascii-ausgleich !! accu+48d
00248 LD (HL),A ; dann wird aus 01h 49h,'A' !!
00249 INC HL
00250 LD A,'.'
00251 LD (HL),A
00252 INC HL
00253 LD A,'.'
00254 LD (HL),A
00255 INC HL
00256 LD A,(dri2) ; destination drive
00257 ADD A,48 ; ascii-ausgleich !!!!!!!!!!!!!
00258 LD (HL),A
00259 INC HL
00260 LD A,0dh ; ENTER kennung für COPY befehl
00261 LD (HL),A ; eintragen
00262 INC HL ; zeigt auf das zeichen nach 0dh
00263 LD (buf1),HL ; buf1 aktualisieren
00264 ; der filename wurde in den buffer eingetragen

```

```

00265 ; es wird eine ebene zurueck gesprungen, ob der
00266 ; nexte eintrag gueltig ist.
00267 zur) POP BC ; die parameter werden
00268 POP DE ; zurueckgesetzt
00269 POP HL
00270 RET
00271 ; die routine ein) f)st den GDOS-Befehl ' ) '
00272 ; also 'COPY' ein.
00273 ; HL zeigt auf den Buffer
00274 ein) LD A,')
00275 LD (HL),A
00276 INC HL
00277 LD A,')
00278 LD (HL),A
00279 INC HL
00280 RET
00281 ;die folgende routine ist der ersatz fuer die 'ein)'
00282 ; in zeile 273 bis 279 falls das betriebssystem
00283 ; NewDOS 80 benutzt wird.
00284 ; zeile 274 bis 280 mussen dann geloescht werden.
00285 ein) ld a,'C'
00286 ; LD (HL),A
00287 ; INC HL
00288 ; LD A,'O'
00289 ; ld (hl),a
00290 ; INC HL
00291 ; ld a,'P'
00292 ; ld (hl),a
00293 ; inc hl
00294 ; ld a,'Y'
00295 ; ld (hl),a
00296 ; inc hl
00297 ; ld a,')
00298 ; ld (hl),a
00299 ; inc hl
00300 ; ret
00301 wei7 LD A,') ; test ob der joker gesetzt ist
00302 CP (HL) ; vergleiche mit fname
00303 JR Z,wei5 ; wenns ein '?' war, ist der
00304 ; buchstabe o.k., ansonsten
00305 JR zur) ; braucht fname nicht weiter
00306 ; getestet werden, er entspricht
00307 ; nicht
00308 ; der geforderten spezifikation
00309 ermi LD A,1 ; im ersten sektor des dir
00310 CALL dirr
00311 LD A,(421fh); an dem byte 1fh befindet sich
00312 ADD A,08h ; die angabe, wieviele fde-
00313 ; sectoren das directory
00314 ; enth)lt.
00315 LD (secanz),A ; speichern
00316 RET
00317 ;
00318 ; Jetzt folgt der eigentliche Kopiervorgang
00319 ; es wird vorher getestet auf anzahl der files = 0
00320 ; und auf **/, also kopiere alles
00321 ; au)erdem wird nach dem letzten eintrag als
00322 ; Endemarkierungsein 'E' eingesetzt
00323 ;
00324 copy LD A,'E' ; Endemarkierung
00325 LD (HL),A ; nach dem letzten befehl
00326 LD A,(filzae) ; keine files kopieren?
00327 CP 00h
00328 JR Z,keifil ; Programm beenden
00329 ; vorher meldung ausgeben
00330 LD A,(4320h) ; erstes zeichen nach

```

```

00331 ; 'KOPIERE'
00332 CP '*') ; falls '**' folgt
00333 JR NZ,soon
00334 LD A,(4321h)
00335 CP ')
00336 JR NZ,soon
00337 LD A,(4322h)
00338 CP '*')
00339 JR NZ,soon
00340 LD HL,meld1
00341 CALL sendms
00342 LD HL,meld2
00343 CALL sendms
00344 CALL setkey
00345 CP ')
00346 JP NZ,dosrdy
00347 LD A,0dh ; noch ein ENTER ausgeben
00348 CALL outch ; auf dem bildschirm ausgeben
00349 soon
00350 keifil LD HL,meld3 ; kein file der source-disk
00351 ; entspricht den
00352 CALL sendms ; geforderten spezifikationen
00353 JP dosrdy
00354 soon LD HL,(buf2) ; anfang des buffers
00355 wei8 CALL sendms ; der copy-befehl wird auf dem
00356 ; bildschirm angezeigt.
00357 LD HL,(buf2)
00358 CALL doscal ; der befehl wird ausgef)hrt
00359 CALL NZ,keinfe ; falls ein fehler aufsetr. ist
00360 LD A,(3840h) ; tastaturzeile mit BREAK
00361 BIT 2,A ; Abbrechen mit <BREAK>-Taste
00362 JR Z,wei9
00363 LD A,39h ; fehler 'Function Aborted'
00364 JP doserr
00365 ; Suchroutine um den n)chsten befehl zu finden
00366 wei9 LD HL,(buf2); anfang des letzten befehls
00367 LD B,0
00368 LD C,9; der n)chste befehl kommt fr)hestens
00369 ADD HL,BC ; nach 9 zeichen.
00370 loop6 LD A,(HL)
00371 CP 0dh ; kommt schon der n)chste?
00372 JR Z,wei10
00373 INC HL ; zeiger ein weiter
00374 JR loop6 ; ohne ende.
00375 wei10 INC HL ; ein hamwa noch
00376 LD A,(HL)
00377 CP ') ; normaler befehl?
00378 JP NZ,dosrdy ; wenn nicht, wars der letzte
00379 LD (buf2),HL ; buf2 zeigt auf den n)chsten
00380 JR wei8 ; befehl und so weiter und so fort
00381 keinfe CP 1ah ; fehlermeldung 'Directory voll'
00382 JR Z,loefil; letzten file wieder loeschen
00383 CP 1bh ; 'Diskette voll' ???
00384 JR Z,loefil
00385 CP 1eh ; 'Directory voll bei erw.' ??
00386 JR Z,loefil
00387 CP 21h ; 'Kein Platz auf diskette' ??
00388 JR Z,loefil
00389 JP doserr
00390 loefil LD HL,4326h; letztes (moesliches)
00391 ; signifikantes
00392 ; zeichen des copy-befehls
00393 LD DE,4329h ; die ganze zeichenkette
00394 ; soll um 3 zeichen nach
00395 ; hinten verschoben werden
00396 LD BC,000dh ; sie ist 14 zeichen lang

```

```

00397 ; und zwar 'Filename/Ext'
00398 ; + <ENTER> + ein blank
00399 ; nach dem '>' !!
00400 ; verschiebepfeil
00401 ; <Mit DEC der Parameter !>
00402 LD HL,kill ; source ist 'KILL'
00403 LD DE,4318h ; Anfangs Befehlsbuffer
00404 LD BC,0005h ; übertrage 5 Zeichen
00405 LDIR ; also los
00406 EX DE,HL ; Austausch HL <> DE
00407 findde INC HL ; suche den Doppelpunkt
00408 ; für Destination
00409 ; Drive Eintrags
00410 LD A,<HL>
00411 CP ':' ; solange bis ':' gefunden
00412 JR NZ,findde ; zeiger auf Drive-#
00413 LD A,<dri2> ; Destination-Drive
00414 ADD A,'0' ; ASCII-Ausgleich
00415 LD <HL>,A ; eintragen
00416 INC HL
00417 LD A,0dh ; <cr> am Schluß eintragen
00418 LD <HL>,A
00419 LD HL,4318h ; Befehlsbuffer des DOS
00420 CALL sendms ; auf Display anzeigen
00421 LD HL,4318h
00422 CALL doscal ; ausführen
00423 JP Z,noerr ; zurück ins DOS
00424 LD HL,meld4 ; ansonsten Fehlermeldung
00425 CALL sendms ; ausgeben
00426 JP dosrdy ; und zurück ins DOS
00427 noerr LD HL,meld5 ; 'Diskette voll' auss.
00428 CALL sendms
00429 JP dosrdy
00430 kill DM ; DOS-BEFEHL KILL
00431 meld1 DM 'Wirklich alle Files kopieren?',0dh
00432 meld2 DM 'Dann Jetzt <J> drücken',03h
00433 meld3 DM 'Keine Files der Spezifikation'
00434 DM 'vorhanden.',0dh
00435 meld4 DM 'Der letzte kopierte File muss'
00436 DM 'noch gelöscht werden.',0dh
00437 meld5 DM 'Diskette voll !!!!!',0dh
00438 buffer EQU $
00439 END start

```

Leider ist mein Drucker noch älter als mein Computer, er  
kennt keine Umlaute.

ä = f  
ö = l  
ü = r  
ß = z



149 30 02 176

## Ein bißchen C, ein bißchen Mathe . . .

Vor einiger Zeit hatte ich die Idee, mein C-System um einige mathematische Routinen zu ergänzen. Mit den vier Grundrechenarten war ich nicht ganz so zufrieden, wie es sich die Entwickler des C-Compilers vielleicht gedacht hatten. Bevor so ein Vorhaben aber zu brauchbaren Algorithmen führt, stellte sich mir das Problem selbige erst einmal ausfindig zu machen. Wer hat schon soetwas als Nichtinformatiker in der Schublade liegen.

Gestecktes Ziel war, die fehlenden Funktionen - immerhin alle trigonometrischen Operationen - mit möglichst schnellen Algorithmen aufzubauen. Auscheiden würden somit schon einmal alle höheren Operationen oberhalb "+" und "-". Es mußte also alles so weit als möglich mit Addition und Subtraktion erledigbar sein - zumindest in den zeitfressenden Berechnungsschleifen. Gemacht, getan. Da mir nichts mit viel Mathematik in gedruckter Form zur Hand gehen konnte, war mein erster Schritt nach einem Buch oder auch mehreren, zu suchen und aus diesen die gesuchten Rechenanweisungen herauszuziehen.

Die Rechenfunktionen basieren alle auf "Unendlichen Zahlenfolgen". Beim "TAN(x)" habe ich auf "SIN(x)/COS(x)" zurückgegriffen, da hier eine Folge zur Berechnung der Bernoulischen Zahlen erforderliche ist. Die Berechnung ist recht aufwendig und deshalb habe ich mir diesen Akt verkniffen. Anzumerken ist noch, das die vorgestellten Routinen noch nicht der "Weisheit letzter Stuß" darstellen. Eine der Algorithmen konvergieren bei bestimmten Winkeln nur sehr schlecht, d.h., sie rechnen und rechnen und .... Zum Teil auch deshalb, weil die Genauigkeit des Fließkomma-Routinen meines C-Compilers nur 6 ... 7 Stellen umfaßt, was, wie ich mit GFA-Basic feststellen konnte, eine der Ursachen ist. Grundsätzlich bestimmt "FLT\_EPSILON" die Genauigkeit der Ergebnisse. Zu beachten ist aber, daß die erreichte Genauigkeit des errechneten Wertes geringer ausfällt als "FLT\_EPSILON" angibt. Bei "FLT\_EPSILON" gleich 1E-6 wird auf 6 Stellen hinter dem Komma gerechnet. Wenn sich dann das (n-1)-te in der Differenz von n-ten Ergebnis um weniger als "FLT\_EPSILON" unterscheidet, wird die Berechnung abgebrochen. Bei 1E-9 ist also nach 9 Stellen Schluß. Damit diese Genauigkeit aber auch erreicht wird, muß die benutzte Fließkomma-Arithmetik mit einer größeren Stellengenauigkeit rechnen als man erreichen will !!! Sonst geht die Sache wegen der unvermeidlichen Rechenfehler schief. Das sicherste ist, die Stellenzahl zwei Stellen weniger als möglich einzustellen. Dann bleibt genügend Luft für Rundungsfehler etc.

Das Buch, aus dem ich alles entnommen habe nun als letztes (S. 603):

### "Rechnen und Mathematik"

Das Lexikon für die Schule und Praxis

Dudenverlag Mannheim/Wien/Zürich  
4. Auflage, ISBN 3-411-02423-2  
Preis: DM 36.--

In dem Schinken steht auch noch einiges mehr. Wer sich in Sachen Algorithmen noch weiter vergraben will, kann es hier ausgiebig tun.

Tschüß

*Rust*

```
/*
 * Die nachfolgenden "#define"-Anweisungen legen einige Konstanten
 * zu den Algorithmen fest
 */

#define M_PI          3.14159265
#define M_PI_2        (M_PI)/2
#define M_LN10        2.30258509
#define FLT_EPSILON   1E-6

/*
 * Funktion zur Berechnung des FMOD(X,Y)
 * Es wird der Gleitkomma-Rest der Modulo-Funktion
 * X/Y berechnet ! Hierzu wird bestimmt, wie
 * oft X ganzzahlig durch Y dividierbar ist und mit
 * dem gefundenen Wert rückgerechnet.
 */
float fmod(x,y)
float x,y;
{
    if (x>=0)
        return (x-(float)((int)(x/y)*y));
    else
        return (x-(float)((int)((x/y)+.5)*y));
}

/*
 * Die Funktion 'frac(x)' liefert den Nachkomma-
 * teil einer gebrochen rationalen Zahl.
 */
float frac(x)
float x;
{
    if (x>=0)
        return (x-(float)((int)(x)));
    else
        return (x-(float)((int)((x)+.5)));
}

/*
 * Funktion zur Berechnung des SIN(X).
 * 'X' ist im Bogenmaß !
 *
 *      x^3   x^5   x^7
 * sin(x)=x - --- + --- - --- + ...   (für x Element von R)
 *           3!   5!   7!
 */
float sin(x)
float x;
{
    register
    float n = 2.,
```

```
        x2 = 0.,
        x_alt = 0.,
        y = 0;

    if(abs(x)>M_PI) x = fmod(x,M_PI);
    x2 = x*x;
    y = x;
    do
    {
        x_alt = y;
        x *= (-x2)/(n*(n+1.));
        y += x;
        n += 2.;
    } while (abs(x_alt-y)>FLT_EPSILON);
    return(y);
}

/*
 * Funktion zur Berechnung des SINH(X).
 * 'X' ist im Bogenmaß !
 *
 *      x^3   x^5   x^7
 * sinh(x)=x + --- + --- + --- + ...   (für x Element von R)
 *           3!   5!   7!
 */
float sinh(x)
float x;
{
    register
    float n = 2.,
          x2 = 0.,
          x_alt = 0.,
          y = 0;

    /* if(x>M_PI_2) x = fmod(x,M_PI_2); */
    x2 = x*x;
    y = x;
    do
    {
        x_alt = y;
        x *= x2/(n*(n+1.));
        y += x;
        n += 2.;
    } while (abs(x_alt-y)>FLT_EPSILON);
    return(y);
}

/*
 * Funktion zur Berechnung des cos(X).
 * 'X' ist im Bogenmaß !
 *
 *      x^2   x^4   x^6
 * cos(x)= 1 - --- + --- - --- + ...   (für x Element von R)
 *           2!   4!   6!
 */
```

```
 */
float cos(x)
float x;
{
    register
    float n = 1.,
          x2 = 0.,
          x_alt = 0.,
          y = 1.;

    if(abs(x)>M_PI) x = fmod(x,M_PI);
    x2 = x*x;
    x = 1.;
    do
    {
        x_alt = y;
        x *= (-x2)/(n*(n+1.));
        y += x;
        n += 2.;
    } while (abs(x_alt-y)>FLT_EPSILON);
    return(y);
}

/*
 * Funktion zur Berechnung des cosh(X).
 * 'X' ist im Bogenmaß !
 *
 *      x^2   x^4   x^6
 * cosh(x)=1 + --- + --- + --- + ...   (für x Element von R)
 *           2!   4!   6!
 */
float cosh(x)
float x;
{
    register
    float n = 1.,
          x2 = 0.,
          x_alt = 0.,
          y = 1.;

    /* if(x>M_PI) x = fmod(x,M_PI); */
    x2 = x*x;
    x = 1.;
    do
    {
        x_alt = y;
        x *= x2/(n*(n+1.));
        y += x;
        n += 2.;
    } while (abs(x_alt-y)>FLT_EPSILON);
    return(y);
}
}
```

```

/*
 * "X" im Bogenmaß
 *
 *
 *  $\text{atan}(x) = x - \frac{1}{3}x^3 + \frac{1}{5}x^5 - \dots + \dots$ 
 *
 */
float tan(x)
float x;
{
    return((abs(x)<0.005) ? x : (sin(x)/cos(x)));
}

float atan(x)
float x;
{
    register
    float n      = 3.,
          x2     = x*x,
          x_alt  = 0.,
          y      = x;

    do
    {
        x_alt = y;
        x    *= (-x2);
        y    += (x/n);
        n    += 2;
    } while (abs(x_alt-y)>FLT_EPSILON);
    return(y);
}

float cot(x)
float x;
{
    return((abs(x)<0.005) ? (1/x) : (cos(x)/sin(x)));
}

/*
 * |X| < 1
 *
 *
 *  $-\ln \frac{1+x}{1-x} = x + \frac{1}{3}x^3 + \frac{1}{5}x^5 + \dots$ 
 *
 */
float ln(x)
float x;
{
    register
    float d = (x-1.)/(x+1.),
          d2= d*d,
          n = 3.,

```

```

    x = d,
    y = 0.;

do
{
    y = x;
    d *= d2;
    x += (d/n);
    n += 2.;
} while (abs(x-y)>FLT_EPSILON);

return(x+x);
}

/*
 * x Element der reellen Zahlen
 *
 *
 *  $\text{exp}(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$ 
 *
 */
float exp(x)
float x;
{
    register
    float n      = 1.,
          x_alt  = 0.,
          y      = 1.,
          z      = 1.;

    do
    {
        x_alt = y;
        z    *= x/n;
        y    += z;
        n    += 1.;
    } while (abs(y-x_alt)>FLT_EPSILON);
    return(y);
}

/*
 * Von der rechnung her wie "EXP(X)", es wird nur entsprechend
 * umgerechnet
 *
 */
float expl0(x)
float x;
{
    register
    float n      = 1.,
          x_alt  = 0.,
          y      = 1.,

```

```
z = 1.;
x *= M_LN10;
do
{
    x_alt = y;
    z    += x/n;
    y    += z;
    n    += 1.;
} while (abs(y-x_alt)>FLT_EPSILON);
return(y);
}

/*
 * Hier ist mir der Zettel verloren gegangen. Pech, aber die
 * Rechnung kann sich jeder selbst rekonstruieren
 */
/*
float sqrt(x)
float x;
{
    register
    float x1 = 1.,
          x2 = x1;
    int i = 0;

    do
    {
        x1=x2;
        x2=(x1+(x/x1))/2;
        i++;
    } while (abs(x1-x2)>FLT_EPSILON);

    return(x2);
}

main()
{
    float x = 8.9;

    printf("Funktionsargument x = %2.2f\n",x,"n");/**/
    printf("FMOD(x,8) = %3.7f\n",fmod(x,8.)); /**/
    printf("FRAC(x) = %3.7f\n",frac(x)); /**/
    printf("SIN(x) = %3.7f\n",sin(x)); /**/
    printf("COS(x) = %3.7f\n",cos(x)); /**/
    printf("SINH(x) = %3.7f\n",sinh(x)); /**/
    printf("COSH(x) = %3.7f\n",cosh(x)); /**/
    printf("TAN(x) = %3.7f\n",tan(x)); /**/
    printf("COT(x) = %3.7f\n",cot(x)); /**/
    printf("ATAN(x) = %3.7f\n",atan(x)); /**/
    printf("LN(x) = %3.7f\n",ln(x)); /**/
    printf("EXP10(x) = %3.7e\n",exp10(x)); /**/
    printf("EXP(x) = %3.7e\n",exp(x)); /**/
    printf("SQRT(x) = %3.7f\n",sqrt(x)); /**/
}
}
```

# ..Hardware ..Hardware

Umrüstung ade !

Da hatte mir Fritz Chwolka vor längerer Zeit mal am Telefon gesagt, er wüßte von einem Knaben, der seinen 8085-Computer umgerüstet hätte auf Z80.

Das war natürlich die Sensation für mich! Ja, ich benütze noch immer diese Kiste mit CPU 8085A und bin es zuweilen auch zufrieden damit ( AlphaTronic P3 !! hi hi ). So bemühte ich mich denn monatelang, bei der von Fritz genannten Adresse anzurufen, bis ich endlich Erfolg hatte. Aus den mir zugesandten Unterlagen konnte ich die Adresse einer Firma rekonstruieren, die mir dann weitere Unterlagen zusandte und mir anbot, mir versuchsshalber eine Umrüstplatte zuzusenden.

Da ich mir der technischen Details noch nicht ganz sicher war, beließ ich es erstmal bei den Unterlagen.

Diesen Unterlagen zufolge handelt es sich um eine Platine mit Baustein NSC 800 (das ist eine Z80-CPU) und einem Chip 74LS240, die anstelle des 8085 eingesteckt wird. Das ist fast alles. Je nach Taktfrequenz sind das dann verschiedene Typvarianten. Bei meinen 6 MHz Takt ( nicht schlecht, was ? ) sollte das Ganze dann ca. 140 DM kosten.

Ich hatte mich schon fast dazu entschlossen, diesen Betrag zu investieren, als ich doch noch mal mein techn. Handbuch des P3 durchblätterte, und feststellte, daß der Drucker über SID/SOD - Leitungen angesteuert wird. Das ist eine eigene kleine serielle Schnittstelle, die die CPU 8085 direkt anbietet.

Und das wars dann auch, denn als Z80-Processor hat der NSC800 diese Leitungen natürlich auch nicht und ich hätte nach der Umrüstung nicht mehr drucken können.

Aus und vorbei! (Änderungen des Betriebssystems und des Bios ziehe ich wegen fehlender Unterlagen gar nicht in Betracht).

Sollte jemand dennoch Interesse an der Sache haben, Fritz bekam die Unterlagen von mir und auch ich gebe gern Auskunft.

Mit freundlichen Grüßen

*Uwe Schobert*

## CPU280 - die Z80-VAX

Ich hatte schon befürchtet, daß ich mich vor Drohbriefen nicht mehr retten kann, wenn ich den 101. Artikel zum Z280 verzapfe, aber bis jetzt scheint (noch) niemand was darüber schreiben zu wollen. Müssen erst die Namen der Besitzer veröffentlicht werden, damit die was von sich hören lassen?

Na gut, dann will ich mal mit gutem Beispiel voran gehen. Worum geht es bei diesem Rechner, über den hin und wieder so geheimnisvolle Andeutungen in diversen Artikeln aufgetaucht sind? Ganz einfach, im Prinzip geht es um einen Computer, der als Prozessor keinen schnöden Z80, sondern einen Z280 hat. Der Z280 ist das, was vor langer Zeit mal der Z800 werden sollte und wo der HD64180 bzw. Z180 auf halbem Weg verhungert ist. Man könnte den Z280 als den 80286 der Zilog-Welt bezeichnen und im Grunde handelt es sich mehr um einen 16-Bit- als um einen 8-Bit-Prozessor. Um nur einige Highlights kurz zu nennen: Rund 600 Befehle, die über den Befehlssatz des Z80 hinausgehen (16-Bit-Multiplikation, 32-Bit-Division, relative Adressen und Sprünge über die gesamten 64K...), 3-stufige Befehlspipeline, eine on-chip MMU, die 16MB verwalten kann, 256 Byte Cache, drei 16-Bit Zähler/Timer, vier DMA-Kanäle und eine serielle Schnittstelle. Von der Leistung her ist der Chip dem 80286 mindestens ebenbürtig, wenn nicht sogar überlegen. Trotzdem ist er 100%ig kompatibel zum Z80 und das macht ihn überhaupt erst wirklich so interessant.

Aber nun zum eigentlichen Rechner, der CPU280 (wer als erster errät, woher der Name kommt, bekommt ein Autogramm). In der "Grundausstattung" besteht der ganze Compi lediglich aus einer Europakarte, die den Prozessor mit 64 oder 128KB ROM und bis zu 4MB (!) dynamischem RAM, den Floppycontroller (kompatibel zum 765), zwei interruptfähige serielle Schnittstellen und eine akkugepufferte Echtzeituhr mit 50 Bytes statischem Ram für Konfigurationsdaten beherbergt. Mit einem Netzteil, einem Terminal und zwei Laufwerken hat man schon einen kompletten CP/M-Rechner zusammen und kann loslegen. Das CP/M wird normalerweise "in Nullzeit" direkt aus dem EPROM gebootet, zu Testzwecken kann man das System aber auch von der Floppy laden. Die Taktrate beträgt im Moment 12.288MHz, was in der Rechenleistung einem Z80 bei etwa 16-20MHz entspricht, je nach Programm. Wenn man (=Stefan Nitschke) mal wirkliche Rechenpower demonstrieren will und dazu echten Z280-Code benutzt, liefert die Karte ein Apfelmännchen mit 720x348 Punkten und 40 Schritten Iterationstiefe in ca. einer halben Minute ab. Damit die Karte mit der alten ECB-Welt kompatibel ist, wird das Businterface mit einem Takt von rund 6MHz betrieben, intern wird alles aber selbstverständlich 16 bittig und ohne Waitzyklen mit voller Geschwindigkeit betan.

Worin unterscheidet sich das Board nun von anderen CP/M-Rechnern? Da bis zu 4MB RAM, im platzsparenden ZIP-Gehäuse, bestückt werden können, hat man beim Vollausbau, wenn man die diversen Puffer abzieht, eine Ramdisk von rund 2-3MB zur Verfügung und da der Z280 hier im 16-Bit-Burst-Modus (4 Worte werden mit 2 Speicherzyklen gelesen) betrieben wird, eine verdammt schnelle. Zum Thema Disketten hat sich Tilmann Reh auch etwas ganz feines einfallen lassen. Man kann auf die Disketten einen Parameterblock schreiben, sodaß der Rechner das Format automatisch erkennen kann. Damit kann man in einem Laufwerk z.B. beliebig zwischen HD-Floppies mit 1.44MB (5

# Hardware

1/4" !) und 40 Track SS SD mit 70KB wechseln, ohne daß man etwas per Hand einstellen muß, wodurch sich dieser Rechner hervorragend als Formatkonverter und Kopierstation eignet. Auf 3 1/2"-Disketten gehen sogar 1.77MB, IBM war da unnötigerweise etwas sehr großzügig mit seinen Formaten.

Den Rechner mit CP/M zu fahren ist eigentlich wie ein Porschemotor in einem Goggo, aber CP/M ist leider nun mal DAS Betriebssystem für den Z80. Wenn man nur CP/M laufen lassen will, kann man sich genauso gut einen anderen Composter zulegen, obwohl man wohl länger suchen muß, bis man für knapp 600 Mark diese Ausstattung auf einer Karte findet. Wer aber Lust am Experimentieren hat, der ist mit diesem Chip bestens bedient und wenn dann noch das Club-Terminal als Ein-/Ausgabe dient, dürfte die CPU280 einer der leistungsfähigsten CP/M-Rechner sein, die es momentan gibt. Natürlich läuft die Version 3.1, die selbstredend alle Features der Karte voll unterstützt, und wer ZCPR kennt, weiß was Z3PLUS auf so einer Maschine bedeutet. In nächster Zukunft gibt es noch eine AT-Bus-Karte, sodaß man auch in Sachen Massenspeicher ganz vorne mitmischen kann. Da läuft einem doch das Wasser im Munde zusammen, oder? Eine der wenigen Alternativen wäre natürlich das Z280-Board für das Model I oder IV, aber ob das jemals kommt? Auf dem nächsten Clubtreffen sind mit Sicherheit wieder mehrere CPU's anzutreffen, dann könnt Ihr mal "Probefahren". Für Anfragen stehe ich natürlich gerne zur Verfügung, wer sich aber lieber gleich an den Entwickler wendet, seine Adresse ist:

Tilmann Reh  
In der Großenbach 46  
5900 Siegen

Alexander Schmid



»Warum gehst du nicht raus, beschäftigst dich ein bißchen im Garten und läßt mich in Ruhe arbeiten?«

## Einfachst A/D- und D/A-Wandler

Es hat auch in unserer Postille schon diverse Bauanleitungen für A/D- und D/A-Wandler gegeben, aber beides zusammen, und dann noch in einem IC, habe ich noch nicht gesehen. Mit noch weniger Chips geht's ja wirklich nicht, hier dürfte die Untergrenze an Bauteilen endgültig erreicht sein.

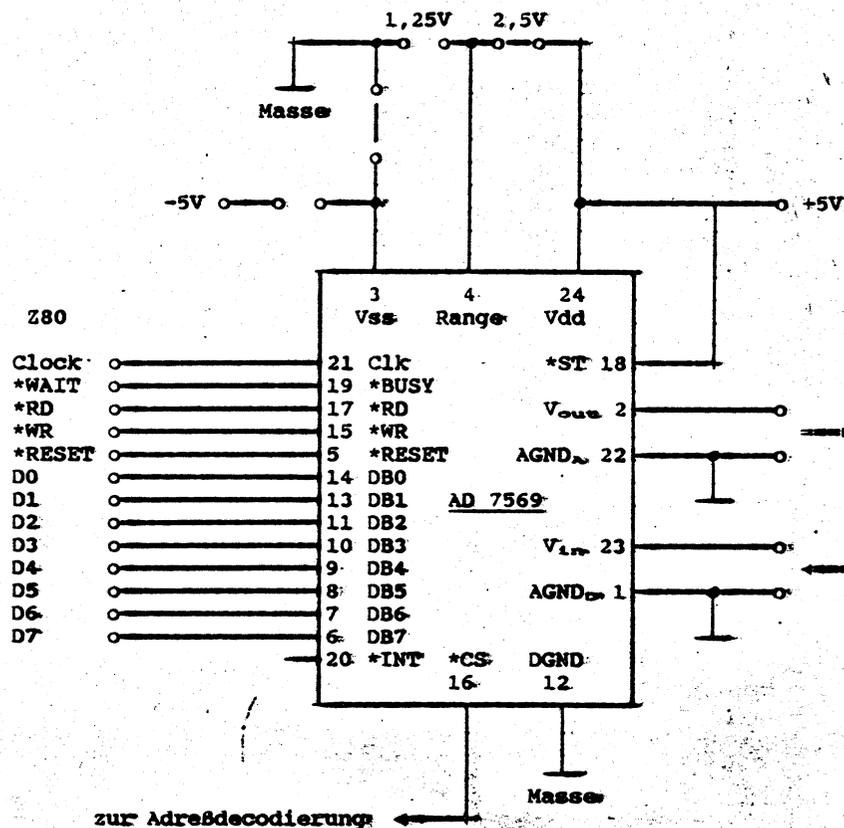
Der AD 7569 von Analog Devices enthält alles was man dazu braucht, einen 8-bit-A/D-Wandler mit 2µs Wandlungszeit, einen D/A-Wandler mit 1µs, die Referenzspannungsquelle und ein Businterface für die direkte Kopplung mit dem Rechnerbus. Bei einer unsymmetrischen Versorgung mit 5V stehen als Ein- und Ausgangsspannungsbereiche 0...1,25V oder 0...2,5V (abhängig vom logischen Pegel am RANGE-Eingang) zur Verfügung. Bei einer symmetrischen Versorgung gehen die Werte entsprechend nochmal genauso weit in den negativen Bereich. Die Adreßdecodierung ist nicht mit auf dem Chip enthalten, je nach Aufwand braucht man dazu nochmal ein oder zwei ICs. Wie sowas geht, könnt Ihr in fast allen Artikeln nachlesen, die sich mit Hardware beschäftigen, ich habe mir das hier erspart, ICs mit dem Texteditor zu zeichnen ist etwas mühsam.

Bei Schreibzugriffen wird der D/A-Wandler aktiv und setzt die Daten, die er auf dem Bus findet, in eine entsprechende Ausgangsspannung um. Bei einem Lesezugriff wird die A/D-Wandlung gestartet, wobei der Chip den Prozessor über die Busy-Leitung solange anhält, bis er fertig ist. Bei einem BASIC-Programm ist aber schon die Bearbeitungszeit der Befehle so lang, daß man davon mit Sicherheit nichts merkt. Auch auf einem 4MHz-Z80 dauert der kürzeste Maschinenbefehl schon 1µs, das als Vergleich. Da lohnt es sich noch nicht mal, die Wandlung zu starten und dann auf einen Interrupt zu warten, während der Rechner etwas anderes macht, das kostet durch den Verwaltungsaufwand viel mehr Zeit. Bei einem 486er mit 33MHz mag das anders aussehen, aber von sowas reden wir hier ja garnicht.

Bei dem Chip von einer nenneswerten Stromaufnahme zu reden ist schon fast lächerlich, gerade mal 12mA braucht er und das kann man wohl jedem Netzteil zusätzlich zumuten. Für den Preis gilt ähnliches, rund 21 Mark sind für diesen Chip sicher nicht zuviel.

Apropos Netzteil, damit kann man z.B. ein billiges programmierbares Netzteil äußerst einfach aufbauen. Über dem D/A-Ausgang steuert der Rechner die Leistungstransistoren direkt an und mit dem A/D-Eingang lauert er, ob an den Klemmen die gewünschte Spannung tatsächlich erreicht wird. Natürlich ist die Regelung etwas träger als eine mit einem integrierten Spannungsregler oder Operationsverstärker, aber wenn die Laständerung nicht zu sprunghaft ist, dürfte es durchaus genügen. Im Gegensatz zu "normalen" Netzteilen, deren Ausgangsspannung immer mehr oder weniger stark lastabhängig ist, kann der Rechner die Spannung dank der Messung (natürlich im Rahmen der 8 Bit) genau auf dem eingestellten Wert halten. Wer sich für (digitale) Regelungstechnik interessiert, könnte hier prima die Wirkungsweise von verschiedenen Regelungsalgorithmen ausprobieren. Auch zum rechnergestützten Aufnehmen von Kennlinien eignet sich sowas gut, denn da braucht man sowieso immer beides, sowohl eine Spannungs- oder Stromquelle als auch ein Meßgerät.

Alexander Schmid



zur Adreßdecodierung ←



# Club 80 Börse --- Club 80 Börse

Genie IIIs, 256 kB RAM, 1,77/7,2  
MHz, 2x Floppy 80/ds/dd (bisher ohne  
Helmut'sche Umbauten)  
Star Gemini 10x 9-Nadel-Drucker (Ep-  
son-Emulation), Centronics-Interface,  
HRG- und download-fähig, NLQ, Trak-  
tor

Zusammen VB 600,- DM, einzeln  
550,- DM (Computer) und 150,- DM  
(Drucker)

Anfragen an Arnulf Sopp, Kathari-  
nenstr. 27, 2400 Lübeck 1 (neue Te-  
lefon- und Fax-Nummer nach Umzug  
noch nicht bekannt).

## "Wegen Geschäftsaufgabe"

verschenke bzw. verkaufe ich folgende Literatur für TRS80/GENIE:

Maschinencode und besseres BASIC (Stewart)	DM 10.--
Machine Language Disk I/O (Wagner)	20.--
Maschinensprachebuch für TRS80 und GENIE (Röckrath)	-----
Z80 Assemblersprache (deutsches Benutzerhandbuch)	20.--
TRS80/Z80 Assembly Language Library (Lindley)	45.--
How to program the Z80 (R.Zaks)	25.--
Disassembled Handbook for TRS80	-----
TRSDOS & DISK BASIC Reference Manual	15.--
Erfolg mit VISICALC (Hergert)	18.--
BASIC FASTER & BETTER (Rosenfelder) - viele nützliche Routinen in BASIC und Maschinensprache	30.--
AIDS III v.1.5 - Handbuch für eine sehr schnelle Datenbank	-----
<i>Versand unfrei oder gegen Portoerstattung (in Briefmarken).</i>	
<i>K.-J.Mühlenbein, Keplerstr.13 * 6940 Weinheim * Tel. 06201/183640</i>	

Suche für NewDos/GDos:

Codecen.CMD & Optimice.CMD  
aus ALCOR-Pascal-Compiler-System

für Grape21:  
von M. Winter  
Treiberprogramm für Genie IIIs

Volker Dose 04343/1357

# Der Rechner mit der Handbremse

Da neben den Tandy's immer mehr "Fremdrechner" den Weg zu uns finden, möchte ich denen auch mal einen Artikel widmen. Die CPC's waren ja schon immer etwas anders als andere Rechner und so ist es auch beim Timing der CPU. Damit der Z80 dem Video-Controller nicht ins Gehege kommt, bekommt er sog. WAIT-Zyklen aufgebremmt, was zu ca. 20% Verlust an Rechenleistung führt; als effektive Taktfrequenz bleiben rund 3.2MHz übrig. Wenn man für Zeitschleifen u.ä. aber die genaue Zahl der Taktzyklen braucht, hilft ein Blick ins Prozessorhandbuch leider nicht weiter, weil die unterschiedlichen Befehle von den Wait-Zyklen unterschiedlich beeinflusst werden und die Zahl der Takte somit mehr oder weniger stark von einem ungebremsten Z80 abweicht. In der c't 9/85 hat sich mal jemand die Mühe gemacht und diese Zeiten bestimmt. Hier nun das Ergebnis, vielleicht kann es der ein oder andere ja brauchen. Die Zahl der Taktzyklen ist angegeben als "ohne Bremse (=anderer Rechner) / mit angezogener Handbremse (=CPC)". Durch dieses merkwürdige System beträgt die Ausführungszeit eines Befehls immer ein vielfaches von 4 Takten und bei längeren Befehlen kommt meistens noch eine "Ehrenrunde" dazu.

## \*\*\*\*\* 8-Bit Ladebefehle \*\*\*\*\*

Befehl	Takte
LD r,r'	4 / 4
LD r,n	7 / 8
LD r,(HL)	7 / 8
LD r,(IX/IY+d)	19 / 20
LD (HL),r	7 / 8
LD (IX/IY+d),r	19 / 20
LD (HL),n	10 / 12
LD (IX/IY+d),n	19 / 24
LD A,(ss)	7 / 8
LD A,(nn)	13 / 16
LD (ss),A	7 / 8
LD (nn),A	13 / 16
LD A,I/R	9 / 12
LD I/R,A	9 / 12

## \*\*\*\*\* 8-Bit Arithmetik \*\*\*\*\*

Befehl	Takte
ADD/ADC/SUB/SBC/ AND/OR/XOR/CP	
-- r --	4 / 4
-- n --	7 / 8
-- (HL) --	7 / 8
-- (IX/IY+d) --	19 / 20
INC/DEC	
-- r --	4 / 4
-- (HL) --	11 / 12
-- (IX/IY+d) --	23 / 24

## \*\*\*\*\* 16-Bit Ladebefehle \*\*\*\*\*

Befehl	Takte
LD dd,nn	10 / 12
LD IX/IY,nn	14 / 16
LD (HL),nn	16 / 20
LD dd,(nn)	20 / 24
LD IX/IY,(nn)	20 / 24
LD (nn),HL	16 / 20
LD (nn),dd	20 / 24
LD (nn),IX/IY	20 / 24
LD SP,HL	6 / 8
LD SP,IX/IY	10 / 12
PUSH qq	11 / 16
PUSH IX/IY	15 / 20
POP qq	10 / 12
POP IX/IY	14 / 16

## \*\*\*\*\* 16-Bit Arithmetik \*\*\*\*\*

Befehl	Takte
ADD HL,dd	11 / 12
ADC HL,dd	15 / 16
SBC HL,dd	15 / 16
ADD IX/IY,dd	15 / 16
INC/DEC dd	6 / 8
INC/DEC IX/IY	10 / 12

## \*\*\*\*\* Exchange, Blocktransfer und Suchbefehle \*\*\*\*\*

Befehl	Takte
EX DE,HL	4 / 4
EX AF,AF'	4 / 4
EXX	4 / 4
EX (SP),HL	19 / 24
EX (SP),IX/IY	23 / 24
LDI/LDD	16 / 20
LDIR/LDDR	
-- B<>0 --	21 / 24
-- B=0 --	16 / 20
CPI/CPD	16 / 16
CPIR/CPDR	
-- B<>0 --	21 / 24
-- B=0 --	16 / 16

## \*\*\*\*\* Sprünge \*\*\*\*\*

Befehl	Takte
JP nn / JP cc,nn	10 / 12
JR e / JR cc,e	
-- cc wahr --	12 / 12
-- cc falsch --	7 / 8
JP (HL)	4 / 4
JP (IX/IY)	8 / 8
DJNZ e	
-- cc wahr --	13 / 16
-- cc falsch --	8 / 12

## \*\*\*\*\* INPUT / OUTPUT \*\*\*\*\*

Befehl	Takte
IN / OUT A,(n)	11 / 12
IN / OUT r,(C)	12 / 16
INI / IND	16 / 20
INIR / INDR	
-- B<>0 --	21 / *
-- B=0 --	16 / *
OUTI / OUTD	16 / 20
OTIR / OTDR	
-- B<>0 --	21 / *
-- B=0 --	16 / *

## \*\*\*\*\* Rotieren und Verschieben \*\*\*\*\*

Befehl	Takte
RLCA/RLA/RRCA/RRA	4 / 4
RL /RLC/RR /RRC/ SLA/SRA/SRL	
-- r --	8 / 8
-- (HL) --	15 / 16
-- (IX/IY+d) --	23 / 28
RLD/RRD	18 / 20

## \*\*\*\*\* CALL und RETURN \*\*\*\*\*

Befehl	Takte
CALL nn / CALL cc,nn	
-- cc wahr --	17 / 20
-- cc falsch --	10 / 12
RET	10 / 12
RET cc	
-- cc wahr --	11 / 16
-- cc falsch --	5 / 8
RETI / RETN	14 / 16
RST p	11 / 16

## \*\*\*\*\* Bit-Verarbeitung \*\*\*\*\*

Befehl	Takte
BIT	
-- b,r --	8 / 8
-- b,(HL) --	12 / 12
-- b,(IX/IY+d) --	20 / 28
SET/RES	
-- b,r --	8 / 8
-- b,(HL) --	15 / 16
-- b,(IX/IY+d) --	23 / 28

\*\*\*\*\* sonstige Befehle \*\*\*\*\*

Befehl	Takte
DAA / CPL / CCF / SCF / NOP / HALT/ EI / DI	4 / 4
NEG / IM 0/1/2	8 / 8

---

r,r': A,B,C,D,E,H,L	b : Bitnummer
dd : BC,DE,HL,SP	p : Restartnummer
ss : BC,DE,HL	d : 8-Bit-Offset
qq : AF,BC,DE,HL	cc : Bedingung
n : Byte oder Portadresse	e : relative 8-Bit-Adresse
nn : 16-Bit-Adresse	

\*) diese Befehle können beim Schneider nicht verwendet werden

Alexander Schmid

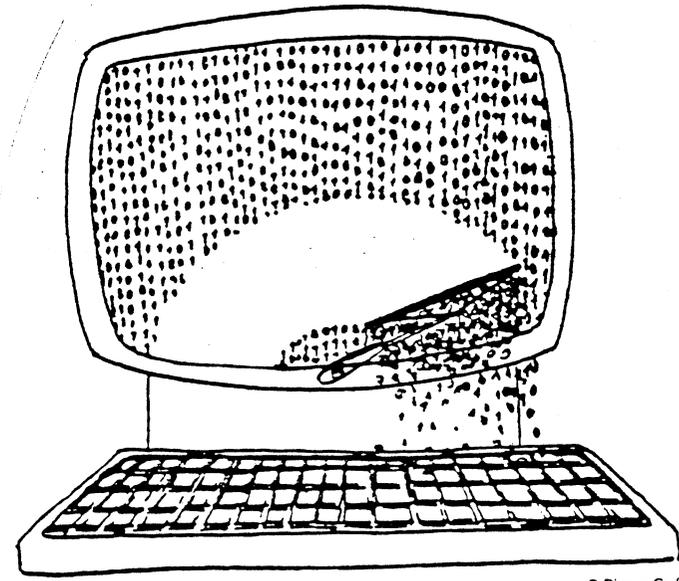
## Schönere Schildchen

Wenn man seinem Rechner eine neue Schnittstelle gönnt oder sonstige Veränderungen vornimmt, kommt man immer wieder zu dem Punkt, wo es darum geht, der Öffentlichkeit kundzutun, wozu dieser und jener Stecker dient. Handgeschriebene Aufkleber haben leider den Nachteil eben genauso auszusehen. Neidvoll blickt man immer wieder auf die professionellen Beschriftungen "ordentlicher" Rechner, aber es gibt einen recht einfachen Weg, das auch selber hinzukriegen. Man nehme selbstklebende Etiketten, wie es sie für fast jeden Drucker gibt und spanne sie in denselben ein. Danach drucke man den gewünschten Text möglichst fett und dunkel, damit die Schrift nicht so nach Nadeldrucker aussieht. Das Wichtigste kommt jetzt: damit nichts mehr verwischen oder verkratzen kann nehme man einen Streifen Tesa o.ä. und klebe ihn auf das Etikett! Mit dem Fingernagel ordentlich festreiben und anschließend kann man das Schildchen mit der Schere in Form bringen und auf den Compi kleben, wenn nötig mit Uhu. Das Ergebnis kann sich durchaus sehen lassen und man braucht seinen Rechner nicht mehr schamvoll zu verstecken. Aufreibebuchstaben sehen zwar auch gut aus, aber um damit eine gerade Linie hinzukriegen muß man schon viel üben, besonders wenn einem Stecker und Knöpfe im Weg sind und billiger ist das sicher nicht. Ganz abgesehen davon, daß diese Zeichen recht empfindlich sind und man sie für den rauen Alltagsbetrieb am besten mit Klarlack schützen sollte.

Club 80  
INFO 36  
Jan. 92

Seite  
33

Alexander Schmid



## Hacker's Hirnkrücke

Ihr werdet Euch sicher wundern, was ein Taschenrechner in dieser Postille macht, aber wenn man PC als "Pocket Calculator" interpretiert, kommt's doch hin. Es geht einfach darum, daß ich endlich einen brauchbaren Hardware-Sidekick gefunden habe, der neben allen von mir benötigten Funktionen auch in Hex rechnen kann. Besagter Rechner ist der CASIO fx-115D und zählt sich zu den sog. wissenschaftlichen Rechnern, d.h. er hat alle möglichen trigonometrischen Funktionen und was man in der höheren Mathematik sonst noch so braucht. Dazu hat er natürlich statistische Funktionen und kann sogar Permutationen und Kombinationen berechnen (wer nicht weiß, was das ist, braucht es auch nicht, wer damit aber schon mal zu tun hatte, weiß, was das für ein Aufwand ist). Daß er mit komplexen Zahlen umgehen kann, ist schon nicht mehr so selbstverständlich und für den Programmierer schließlich bietet er beliebige Umrechnungen zwischen hex, oktal, binär und dezimal. Als logische Funktionen hat er NEG (2er-Komplement), NOT (1er-Komplement), AND, OR, XOR und XNOR, dazu kommen selbstverständlich noch die arithmetischen Funktionen (+, -, \*, /) in allen Zahlensystemen (was war doch gleich 123h \* 4...). Mehr kann man wohl nicht verlangen. Ansonsten ist es ein ganz normaler Taschenrechner, wie es ihn sicher auch von anderen Firmen gibt, fragt sich nur, ob man dort auch so viele Funktionen zusammen hat. Programmierbare Rechner von HP oder Texas waren mir bis jetzt immer eine Nummer zu groß und zu teuer, den CASIO gibt's schon für 35 Mark. Warum soll man mit Kanonen auf Spatzen schießen und mehrere hundert Mark ausgeben, etwas selber denken tut hin und wieder auch ganz gut.

Club 80  
INFO 36  
Jan. 92

Seite  
34

Alexander Schmid

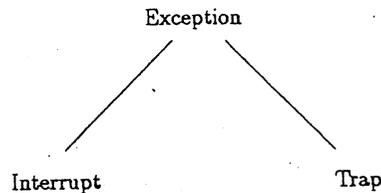
Dieser Artikel fällt wieder in das Themengebiet „Betriebssysteme“, wie sollte es anders sein. Die drei obengenannten Begriffe sind auf vielen Ebenen zu finden, von der Maschinenebene (Mikroprozessor und Konsorten) bis zur Ebene der Programmiersprachen (PL/1 und Nachfolger). Die ganze Sache ist für ProgrammiererInnen interessant, die sich nahe an der Hardware tummeln, sei es in Assembler oder einer (höheren) Programmiersprache. Übrigens finden sich die Begriffe bei *allen* Prozessortypen (selbst bei den Intel-80x86), obwohl der Z80 natürlich alles am schönsten, besten, einfachsten macht...

Leider werden die Begriffe nicht einheitlich verwendet. Ich werde hier *meine* Auffassung der Begriffe darlegen, da sich die verschiedenen Auslegungen nicht unter einen Hut bringen lassen. Deshalb kann es sein, daß Euch die Begriffe in einer etwas anderen Verkleidung über den Weg laufen, aber erkennen und einordnen könnt Ihr sie dann hoffentlich trotzdem.

Ich werde zuerst die Begriffe so vorstellen, wie ich sie verstehe, und dann Einsatzbereiche nennen.

## Exception

Eine Exception (dt.: Ausnahme) ist für mich entweder ein Interrupt oder ein Trap, also ein Oberbegriff:



Der Begriff „Exception“ wird oft im Bereich von höheren Programmiersprachen gebraucht, z. B. bei Ada<sup>1</sup>.

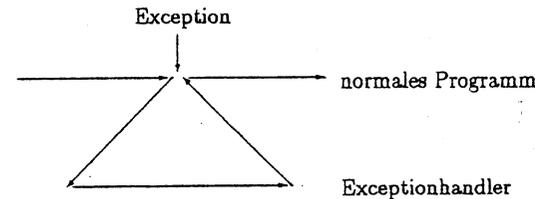
Eine Exception tritt auf, wenn irgendein *Ereignis* den normalen Programmfluß *unterbricht*. Denkbar sind z. B. folgende Ereignisse und Reaktionen:

<sup>1</sup>In C bzw. UNIX heißen Exceptions „Signale“, sind aber ähnlich.

Ereignis	mögliche Reaktion
ein unzulässiger Wert wurde eingegeben	Wiederholung der Eingabe
ein Sektor soll eingelesen werden	der Sektor wird gelesen
Division durch 0	Ausgabe einer Fehlermeldung
Überschreiten der Indexgrenzen eines Feldes	Erweiterung des Feldes
es wird auf einen Speicherbereich zugegriffen, der gerade ausgelagert ist	Einlagerung des Speicherbereichs
ein Zeichen ist an der seriellen Schnittstelle angekommen	das Zeichen wird abgeholt
beim Lesen eines Sektors von der Diskette oder Festplatte ist ein Fehler aufgetreten	das Lesen wird wiederholt und ggf. eine Fehlermeldung ausgegeben
ein Sektor wurde von der Diskette oder Festplatte in einen Buffer (dt.: Puffer, Zwischenspeicher) gelesen	der Sektorinhalt wird aus dem Buffer abgeholt
ein Timer (dt.: Zähler) ist abgelaufen	ein Weckton wird ausgegeben

Alle diese Ereignisse finden während des *normalen* Programmablaufs statt, sollen aber *sofort* bearbeitet werden. Also wird das Programm unterbrochen und ein Unterprogramm aufgerufen, das sich um die *Behandlung der Exception* kümmert; dieses Unterprogramm nennen wir *ExceptionHandler*. Nach der Behandlung der Exception wird das Programm an der Stelle fortgesetzt, an der es unterbrochen wurde.

Der Programmfluß sieht grafisch folgendermaßen aus:



## Interrupt

Ein Interrupt (dt.: Unterbrechung) liegt vor, wenn ein Gerät bzw. ein Controller die Arbeit des Prozessors unterbricht. Ein Interrupt wird also *extern* verursacht und, dann *intern* (im Prozessor) behandelt.<sup>2</sup> Die Behandlung wird von einem *Interrupt-*

<sup>2</sup>Heute finden wir *intern* in einem Mikroprozessor auch Einheiten vor, die früher *extern* (als eigene Chips) realisiert wurden, wie serielle Schnittstellen, DMA-Kanäle (DMA: „Direct Memory Access“, dt.: direkter Speicherzugriff) oder Timer. *Logisch* sind diese Einheiten aber weiterhin *externe* Einheiten, denn der Prozessor würde auch ohne sie funktionieren. Sprich: Interne Einheiten sind solche, ohne die ein Prozessor nicht funktioniert (kein Programm abarbeiten kann).

handler (oft auch „Interruptroutine“ genannt) erledigt. (Siehe dazu auch Info 31, Artikel „Schnittstellenprogrammierung“.)

## Trap

Bei einem Interrupt ist ein *externes* Ereignis dafür verantwortlich, daß der Programmfluß unterbrochen wird. Wenn ein *internes* Ereignis die Ursache für eine Unterbrechung ist, nennen wir die Unterbrechung Trap (dt.: Falle). Für die Behandlung ist ein Traphandler zuständig.

Interne Ereignisse werden durch einen *Befehl* direkt oder indirekt ausgelöst. Eine *direkte* Auslösung ist ein spezieller Trap-Befehl, der dann wie ein Unterprogrammaufruf CALL *xxxx* wirkt, wobei *xxxx* die Adresse des Traphandlers ist. *Indirekt* wird ein Trap ausgelöst, wenn dies bei der Ausführung eines Befehls geschieht (der kein Trap-Befehl ist). Beispiele:

- Beim Versuch, durch 0 zu teilen (was nicht möglich ist), wird ein Trap „Division durch 0“ ausgelöst, was die Ausgabe einer Fehlermeldung zur Folge haben kann.
- Falls ein unbekannter Befehl ausgeführt werden soll, wird ein Trap „Unbekannter Befehl“ ausgelöst, so daß der Traphandler diesen Befehl evtl. emulieren kann (siehe Z80-Illegals, Info 17).

## Einsatzbereiche

Wo lassen sich Exceptions, also Interrupts und Traps, sinnvoll einsetzen?

### Interrupts

Die Einsatzbereiche für Interrupts wurden schon im Artikel „Schnittstellenprogrammierung“ in Info Nr. 31 beschrieben: Wenn der Prozessor „ziemlich lange“ warten muß, weil ein Gerät oder Controller zu *langsam* ist (z. B. Drucker), oder wenn sich nicht voraussehen läßt, *wann* etwas passiert (z. B. Empfang eines Zeichens an der seriellen Schnittstelle), lassen sich Interrupts einsetzen. In der zwischen den Interrupts liegenden Zeit kann der Prozessor einen Haufen „nützlicher“ Arbeit verrichten, statt in Warteschleifen zu kreisen.

### Traps

Trap-Befehle bieten sich als Schnittstelle zwischen Anwenderprogramm und Betriebssystem an. Im Gegensatz zu einem normalen Unterprogrammaufruf per

CALL *xxxx* sind Traps meistens schneller und nicht von einer bestimmten Adresse *xxxx* abhängig, also änderungsfreundlicher. Andere Traps erhöhen vor allem den Komfort und die Sicherheit von Programmen, denn es muß z. B. nicht vor jeder Division  $x \text{ DIV } y$  geprüft werden, ob der Divisor  $y = 0$  ist.

### Weitere Vorteile

Da die Aufrufe der Exception- (Interrupt-, Trap-) Handler meistens nicht direkt, sondern über eine Tabelle erfolgen, in der die Adressen der Handler eingetragen sind, lassen sich leicht neue Handler für einen Interrupt oder Trap installieren: In der Tabelle wird die Adresse des alten Handlers durch die Adresse des neuen Handlers ersetzt. Wenn die Tabelle groß genug ist, lassen sich auch leicht neue Exceptions einführen.

### Alles klar?

Ich hoffe, daß Ihr mit mir der Meinung seid, daß Exceptions eine tolle Sache sind und viel mehr ausgenutzt werden sollten.

Ich muß an dieser Stelle nochmal darauf hinweisen, daß die Begriffsbildung nicht einheitlich ist. Bei den 680x0 werden auch Interrupts als „Traps“ bezeichnet, bei den 80x86 ist es umgekehrt (der Befehl INT ist ein Trap, kein „Interrupt“). Nachdem ich jetzt die Sache ein für allemal klargestellt habe, sollten sich so ~~um~~ bedeutende Firmen wie Motorola (680x0) und Intel (80x86) gefälligst an meine Begriffe halten, meine ich!

# Vom Single zum Multi

Gerald Schröder

## Um was geht's?

Neuerdings taucht in viele Artikeln, Anzeigen und Gesprächen das Schlagwort „Multi“ auf: Multitasking, Multiprocessor, Multiprocessing, Multiuser. Diese Begriffe werden uns alten CP/MLern an den Kopf geworfen, mit dem Vorwurf, ein „Singleprocessor, Singleuser, Singletasking System“ sei doch wohl nicht mehr zeitgemäß.

Ich möchte hier versuchen, die Brücke vom alten „Single“ zum modernen „Multi“ zu schlagen. Dazu werde ich (bei ausreichender Ausdauer) einige Artikel zu diesem Thema schreiben. Anfangen möchte ich hier mit einer Einführung in die obengenannten neuen Begriffe. Entferntes Ziel dieser Serie ist es, den Kern eines modernen Betriebssystems auf Basis des Z280 zu entwerfen. Ich hoffe, daß der Weg dorthin Euch hilft, den ganzen Humbug zu durchschauen, damit Ihr aus vollem Herzen sagen könnt: „So ein Quatsch!“ oder „Stark! Super! Genial!“, je nach Geschmack.

Damit sowohl die englisch- als auch die deutschsprachigen Leser zu ihrem Recht kommen, habe ich ein Glossar englischer Begriffe, die in diesem Text auftauchen, erstellt und als Anhang beigelegt. Es läßt sich darüber streiten, ob eher die englischen Originale, die „Eindeutschungen“ oder die Übersetzungen benutzt werden sollen.

Für die Wissensdurstigen habe ich als zweiten Anhang einige Literaturhinweise verfaßt, die allerdings nur einen minimalen Ausschnitt aus der unübersehbaren Masse an Büchern zu diesem Thema darstellen.

## Prozeß und Prozessor

Was ein Prozessor ist, weiß hoffentlich jeder. Sagen wir einfach: Ein Prozessor ist ein Ding, das Befehle ausführt und Daten bearbeitet. Aber was ist ein Prozeß? Eine mögliche Antwort ist: Ein Prozeß ist ein Programm, wie wir es alle kennen. Genauer ist folgende Definition:

Ein Prozeß besteht aus drei Teilen:

1. Einer Folge von Befehlen.
2. Einem Satz von Daten.
3. Einem inneren und einem äußeren Zustand.

Was Befehle und was Daten sind, dürfte klar sein. Für uns ist nur interessant, daß der Prozessor die Befehle ausführt und mit den Daten arbeitet, insbesondere sie ändert.<sup>1</sup>

Der *innere* Zustand beschreibt den *Abarbeitungsstand* des Prozesses:

- Zustand der Daten.
- Befehlszeiger (Zeiger auf den nächsten auszuführenden Befehl).

Von *außen* gesehen, können wir auch mehrere Zustände eines Prozesses unterscheiden, z. B.:

inaktiv: Der Prozeß wurde noch nicht gestartet.

laufend: Der Prozeß läuft gerade.

wartend: Der Prozeß hat sich selbst in einen Wartezustand versetzt (und wartet auf ein Signal von außen).

unterbrochen: Der Prozeß würde von außen unterbrochen (und wartet auf seine Fortsetzung).

Damit ein Prozeß laufen kann, muß er einen Prozessor zugeteilt bekommen, der die Befehle des Prozesses ausführt. Dadurch wiederum wird der *innere* Zustand verändert, denn erstens ändern die Befehle normalerweise die Daten und zweitens zeigt nach Abarbeitung eines Befehls der Befehlszeiger auf den nächsten Befehl.

## Multitasking bzw. Multiprocessing

Der Normalzustand, den wir alle aus der täglichen Praxis kennen, ist *ein* Prozeß auf *einem* Prozessor. Sobald wir es mit *mehreren* Prozessen zu tun haben, gibt es mehrere Möglichkeiten der Ausführung:

1. Wir lassen die Prozesse *nacheinander* auf *einem* Prozessor laufen. Diese Möglichkeit ist trivial und interessiert uns nicht weiter.
2. Wir lassen die Prozesse *konkurrent* (bzw. nebenläufig) auf *einem* oder *mehreren* Prozessoren ablaufen. Zwei Prozesse sind konkurrent, wenn zu einem Zeitpunkt beide begonnen, aber keiner abgeschlossen wurde [Gil81, Seite 49]. Nun gibt es wieder zwei Unterklassen:
  - (a) Bei *einem* Prozessor ist zu einem Zeitpunkt immer genau *ein* Prozeß aktiv, während alle anderen entweder unterbrochen sind oder freiwillig warten. Ich nenne dies *quasi parallel*.

<sup>1</sup>Befehle dürfen normalerweise nicht geändert werden. Selbstmodifizierender Code ist also tabu.

- (b) Bei mehreren Prozessoren können mehrere Prozesse gleichzeitig aktiv sein. Zwei Prozesse sind *echt parallel*, wenn zu einem Zeitpunkt beide Prozesse aktiv sind.

Was ist genau der Unterschied zwischen *quasi parallel* und *echt parallel*? Nehmen wir an, wir haben es mit zwei Prozessen zu tun, die ausgeführt werden sollen. Im *quasi parallelen* Fall werden einige Befehle des ersten Prozesses ausgeführt und dieser dann von außen unterbrochen. Nun werden einige Befehle des zweiten Prozesses ausgeführt, bis dieser unterbrochen wird. Jetzt kommt wieder Prozeß 1 an die Reihe usw. Im *echt parallelen* Fall dagegen werden die Befehle eines Prozesses auf einem Prozessor und die Befehle des anderen Prozesses *gleichzeitig* auf einem anderen Prozessor ausgeführt.

Das quasi oder echt parallele Ausführen von Prozessen nennt sich Multitasking oder Multiprocessing. Interessanterweise bieten die meisten modernen Betriebssysteme (z. B. UNIX, OS/2) Multitasking an, aber nur auf *einem* Prozessor. Mit mehreren Prozessoren kann kaum ein Betriebssystem umgehen. Hier gibt es noch viel Entwicklungsarbeit zu tun.

### Threads

In letzter Zeit tauchen neuerdings Threads auf. Threads sind eine Abart der Prozesse. Bei Prozessen gibt es das Problem, daß (normalerweise) jeder Prozeß für sich abgeschlossen ist, d. h. zwei Prozesse sich nicht dieselben Befehle oder Daten teilen dürfen. Wenn zwei Prozesse Daten austauschen oder mit denselben Daten arbeiten wollen, müssen dafür besondere Maßnahmen getroffen werden, indem zum Beispiel ein Prozeß einem anderen Daten schickt (was wiederum Zeit kostet).

Bei Threads ist dies genau umgekehrt: Threads teilen sich sowohl Befehle als auch Daten. Allerdings benutzt jeder Thread einen eigenen Stack. Außerdem besitzt jeder Thread seinen eigenen Zustand, wobei der Zustand der Daten für alle Threads gleich ist.

Ansonsten gelten dieselben Aussagen wie für Prozesse: Auch Threads können konkurrenz ausgeführt werden, wobei wir wieder zwischen der quasi und der echt parallelen Ausführung unterscheiden.

Threads sind normalerweise den Prozessen untergeordnet. D. h. ein Prozeß kann aus mehreren Threads bestehen, aber nicht umgekehrt. Es bietet sich zum Beispiel an, mehrere Prozesse auf mehreren Prozessoren *echt parallel* ablaufen zu lassen. Diese Prozesse können dann wiederum aus mehreren Threads bestehen, die quasi parallel auf dem einen Prozessor laufen.

### Multiprozessorsysteme

Bisher haben wir uns nur mit der Software-Seite beschäftigt. Und was gibt es auf der Hardware-Seite? Wir kennen alle unsere Rechner mit *einem Hauptprozessor*.

Neu und interessant sind Rechner mit *mehreren Prozessoren*, Multiprozessorsysteme genannt.

Dazu wieder einige Klarstellungen. Ein Prozessor eines Multiprozessorsystems sollte auch *allein* in einem Rechner arbeiten können, also macht ein Arithmetik-Coprozessor oder ein Grafikprozessor noch kein Multiprozessorsystem. Es gibt aber höchst unterschiedliche Multiprozessorsysteme, die wir nach folgenden Merkmalen klassifizieren können [Gil81, Kapitel 10]:

#### Kopplung:

starke: Die Prozessoren sind über einen *gemeinsamen Speicher* (Shared Memory) verbunden.

schwache: Die Prozessoren können nur Nachrichten (Messages) austauschen, z. B. über einen Bus.

#### Kontrolle:

zentral: Ein Prozeß übernimmt die Kontrolle über alle anderen Prozesse und die Prozessoren.

verteilt: Kein Prozeß hat die Kontrolle.

#### Hardware:

homogen: Alle Prozessoren sind gleich, z. B. nur Z280-Prozessoren.

inhomogen: Unterschiedliche Prozessoren, z. B. Z280 neben 80386.

#### Symmetrie:

symmetrisch: Alle Prozessoren spielen dieselbe Rolle, können also jede Funktion im System übernehmen.

asymmetrisch: Die Prozessoren haben verschiedene Rollen, z. B. kümmert sich ein Prozessor um die Ausgabe auf Drucker, während ein anderer die Ein-/Ausgabe von/auf Festplatten erledigt.

Daraus kann sich nun jeder sein Multiprozessorsystem zusammenbasteln (Gruß an Helmut), z. B. ein System,

- in dem mehrere Z280 werkeln,
- die alle einen eigenen (lokalen) Speicher besitzen und Nachrichten über eine serielle Schnittstelle austauschen,

- wobei alle Prozessoren dieselbe Rolle spielen (also alle Zugriff auf Drucker, Tastatur, Bildschirm usw. haben) und
- ein Prozeß (auf irgendeinem Prozessor) alles kontrolliert,

nennt sich ein „homogenes, schwach gekoppeltes, symmetrisches Multiprozessorsystem mit zentraler Kontrolle“.

## Multiuser

Der Schritt vom mehreren Prozessen zu mehreren Benutzern ist nicht mehr groß. Im normalen Fall sitzt ein Benutzer (User) vor dem Rechner. Es gibt dann meistens einen Prozeß, der Tastatur und Bildschirm dieses Benutzers überwacht. Wenn wir nun noch eine Tastatur und einen Bildschirm an den Rechner anhängen und einen zweiten Prozeß erzeugen, der diese kontrolliert, haben wir ein Multiusersystem. Es gibt zwei Prozesse, also haben wir es mit Multiprocessing tun.

Unser Multiusersystem kann nur einen Prozessor haben, was natürlich bedeutet, daß sich die Benutzer diesen *teilen* müssen, was den englischen Begriff „time sharing“ geprägt hat (die *Zeit* des Prozessors wird *aufgeteilt*). Wenn mehrere Prozessoren vorhanden sind, können die Prozesse auf diese verteilt werden.

## Schluß

Das soll für heute reichen. Wir haben jetzt die grundlegenden Begriffe im Zusammenhang mit „Multi...“ kennengelernt. Sicherlich gibt es Fragen, Unklarheiten oder Wünsche für weitere Erklärungen. Schickt sie mir! Ich freue mich über jede Reaktion.

## Anhänge

### Glossar

Task (Aufgabe): Wie → Process.

Process (Prozeß): Ein Programm, bestehend aus Befehlen, Daten und einem (inneren und äußeren) Zustand.

Thread (Faden): Ein Teil eines Prozesses (→ Process), der mit anderen Threads Befehle und Daten des Prozesses teilt.

Multitaskingsystem : Wie → Multiprocessingsystem.

Multiprocessingsystem (Mehrprozeßsystem): Mehrere → Prozesse laufen auf einem Rechner.

Multiprocessorsystem (Mehrprozessorsystem): Mehrere Prozessoren in einem Rechner.

Multiusersystem (Mehrbenutzersystem): Mehrere Benutzer teilen sich einen Rechner.

Shared Memory (geteilter Speicher): Wie → Global Memory.

Global Memory (globaler Speicher): Speicher, auf den mehrere Prozessoren zugreifen können. Siehe auch → Local Memory.

Local Memory (lokaler Speicher): Speicher, auf den nur ein Prozessor zugreifen kann. Siehe auch → Global Memory.

## Literatur

Wer sich näher mit diesem Thema befassen will, dem sei hier etwas Literatur empfohlen.

Von Seiten der Hardware werden die modernen Rechnerarchitekturen in [Gil81] beleuchtet. Das Buch ist 10 Jahre alt und dementsprechend sind einige der vorgestellten Rechner längst in der Rumpelkammer verschwunden. Trotzdem sind die grundlegenden Konzepte weiterhin gültig. Ich habe mich hier weitgehend an dieses Buch gehalten.

Bei den Einführungen in Betriebssysteme finden sich viele Bücher, die dieses Thema behandeln, allerdings auf einer sehr allgemeinen Ebene. Konkrete Implementierungen werden zumeist im Bereich von UNIX beschrieben, z. B. [ea].

Den theoretischen Hintergrund beleuchten Jessen und Valk in [JV87]. Sie entwickeln Modelle, mit denen Rechensysteme (Hardware und Software) beschrieben werden können, z. B. Petri-Netze. Außerdem werden Probleme und Lösungen in diesem Bereich besprochen, z. B. die Synchronisation von Prozessen oder Bedienstrategien, d. h. wie quasi-parallele Prozesse auf einem System (mit einem oder mehreren Prozessoren) abgearbeitet werden können. Das Buch kann ich nur Leuten mit soliden mathematischen Kenntnissen empfehlen.

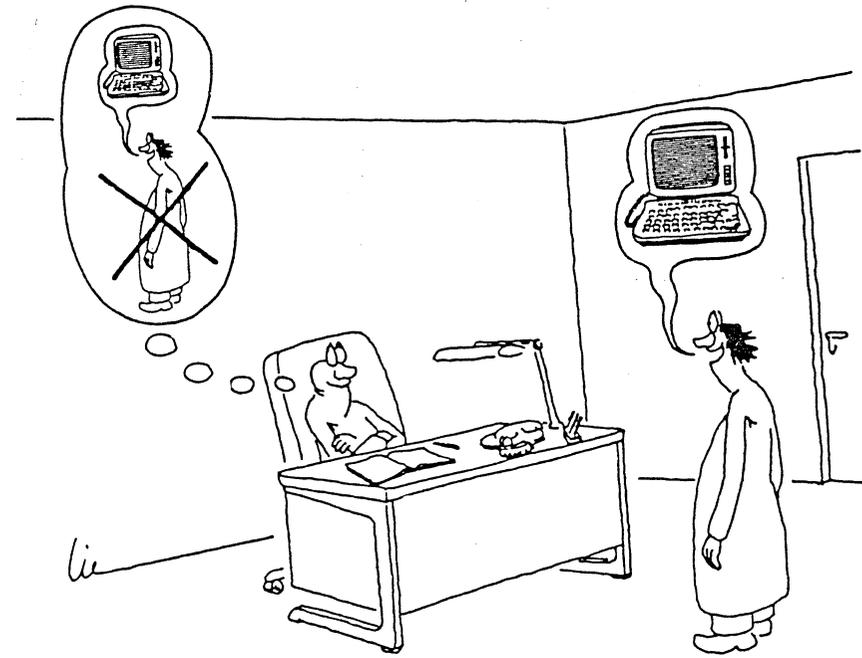
Wer lieber alles zu den den klassischen Themen der Informatik aus einer Hand haben möchte, dem sei A. S. Tanenbaum empfohlen. In [Tan87] bzw. [Tan90a] befaßt er sich mit Betriebssystemen, sowohl allgemein als auch mit einer konkreten Implementierung (MINIX, ein UNIX-Clone, Quelltexte in C werden mitgeliefert). [Tan90b] beschreibt die Hardware von Rechnern näher, setzt aber auf einer tieferen Ebene an als das oben angeführte Buch von Giloi. [Tan88] widmet sich der Verbindung zwischen Rechnern auf den verschiedenen Ebenen. Neben MINIX hat Tanenbaum auch noch an einem verteilten Betriebssystem namens AMOEBA mitgearbeitet, zu dem meines Wissens leider bisher nur Arbeitsberichte erschienen sind.

Mit einer Empfehlung für besondere Zeitschriften kann ich nicht dienen. In fast jeder Zeitschrift findet sich etwas zu dem Thema. Einiges wird z. B. in dem Artikel

[BR91] angesprochen, der auch zeigt, daß dieses Thema noch aktueller Forschungsgegenstand ist.

## Literatur

- [BR91] T. Bemmerl und B. Ries. *Obenliegend. Integrierter Einsatz von Parallelrechnern an der TU München*. iX, 1991(7):74-79, Juli 1991.
- [ea] S. J. Leffler et al. *The Design and Implementation of the 4.3 BSD UNIX Operating System*. ~~2, 27 Seiten~~. Addison-Wesley, 1988. 471 Seiten.
- [Gil81] W. K. Giloi. *Rechnerarchitektur*. Springer, 1981. 380 Seiten.
- [JV87] E. Jessen und R. Valk. *Rechensysteme — Grundlagen der Modellbildung*. Springer, 1987. 560 Seiten.
- [Tan87] A. S. Tanenbaum. *Operating Systems — Design and Implementation, 3rd Ed.* Prentice Hall, 1987. 719 Seiten.
- [Tan88] A. S. Tanenbaum. *Computer Networks, 2nd Ed.* Prentice Hall, 1988. 658 Seiten.
- [Tan90a] A. S. Tanenbaum. *Betriebssysteme — Entwurf und Realisierung (2 Bände)*. Hanser, 1990. 322 bzw. 268 Seiten.
- [Tan90b] A. S. Tanenbaum. *Structured Computer Organization, 3rd Ed.* Prentice Hall, 1990. 587 Seiten.



© Erik Liebermann



© Heinz Langer

## XT = Extra viel Trouble ???

Anfang 1990 hatte ich mich in die Idee verrannt, einen PC200 der Fa. "Sir Sinclair" mit einer Festplatte auszurüsten. Ein einige Zeit zuvor gelesener Testbericht über XT's (der PC200 war ebenfalls getestet worden und bekam eine akzeptable Kompatibilität bescheinigt) unterstützte mich in der Ansicht, daß das Unterfangen durchführbar sei. Also wurde beschlossen, eine 20MB-Platte nebst ST11M-Controller zu beschaffen und beides in der Kiste zu installieren.

Das Wörtchen "akzeptabel" hätte mich hellhörig machen sollen, denn was mit dem Kauf der Platte an Ereignissen losgetreten wurde, reicht aus, um einem den Spaß an der Computerei zu verderben. Da eine Stromversorgung der Platte aus dem Rechner nicht möglich war, mußte ein VERO-Schaltenteil erhalten (klappt übrigens prima). Nach Einsetzen der Karte, Anstöpseln der Stromversorgung und Anbringen aller Kabel konnte es losgehen. Rechner einschalten, Platte auf Drehzahl hochlaufen lassen und auf gehts in schnelle HD-Vergnügen ... Doch back to the roots ! Ohne Formatierung geht's nicht. Das Bios ist so schlau und läßt mir keine andere Wahl. Gemacht, getan !

Ab hier wird die Sache nun skuril. Mit Abschluß der sogenannten Low-Level-Formatierung verabschiedet sich das ganze Vergnügen mit der Meldung "Error: Faulty Memory (Parity-Error)". Von diesem letzten Statement ist die Kiste dann auch nicht mehr abzubringen. Jeder Power-On stoppt an dieser Stelle. Man ist ja Kummer gewohnt und hat gelernt, mit diesen Dingen zu leben.

Das nächste zu erreichende "Opfer" ist der Händler, der mir die Hardware verkauft hat. Auf mein Drängen hin erhalte ich von ihm die Platte für die sofortige Benutzung formatiert, partitioniert und mit dem DOS zum booten beschrieben. Erster Gedanke zu Hause vor dem Rechner: Jetzt muß ja alles bis zu DOS-Prompt in einem Rutsch laufen. Also Controller einstöpseln, Platten & Stromversorgung anschließen und alles einschalten. Der Selbsttest des Rechners läuft an und währenddessen leiert die Platte hoch. Die BIOS-Meldung kommt und ... "Error: Faulty Memory (Parity-Error)". Mittlerweile dämmert es mir, daß das Problem wohl größeren Kalibers ist, denn ziehe ich das Kabel mit den Steuersignalen ab - die Platte wird dann nicht gefunden - läuft alles sauber durch. Es hat den Anschein, als wenn der Rechner mit der Karte so recht nicht will.

Zu denken gibt nur, daß mir der Händler beide Teile in einem AT einwandfrei funktionierend vorgeführt hat. Der Gedanke ist also naheliegend, daß meinem XT nicht so kompatibel wie notwendig ist. Um das abtesten zu können, verabrede ich kurzerhand ein Treffen mit Helmut Bernhard, er besitzt ebenfalls einen XT. Bei ihm eingesteckt läuft alles so wie es soll. Ein vorübergehend ausgeliehener WD1006-Controller offenbart eine neue Fehlermeldung "...controller malfunction...". An ein ordnungsgemäßes Booten des Rechner ist nicht zu denken. Da nun insgesamt zwei Karten in meinem Unikum den Dienst verweigern, sehe ich mich zu einem Brief an den Hersteller genötigt. Da der in England seinen Sitz hat radebreche ich mein Problem in "bestem" Englisch zu Papier und schicke alles auf die Reise. Nach drei Wochen kommt dann endlich die Antwort: Man könne sich meine Schwierigkeiten nicht erklären, der Rechner sei vollkommen "kompatibel", müsse also mit der übliche

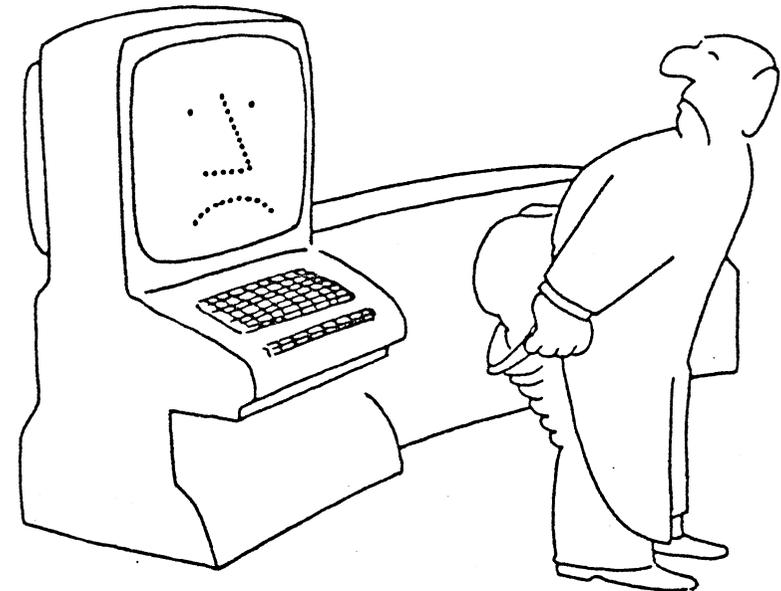
PC-Hardware laufen. Möglicherweise liegt ein Fehler im Netzteil vor (da die Platte extern versorgt wird, ist die Richtigkeit dieser Diagnose fraglich). Ich sollte mich an die deutsche AMSTARD-Vertretung zwecks Unterstützung wenden. Freundlicherweise liefert man im Antwortschreiben auch gleich die Anschrift und Telefonnummer mit. Ein entsprechendes Telefonat beschert mir eine reichlich niederschmetternde Antwort: Da der Rechner hier in Deutschland nicht verkauft wurde, könne man auch keinen Support gewähren, im übrigen hätte man auch keine Serviceunterlagen zur Hand und könne schon aus diesem Grunde nicht helfen (Gott sei Dank ist der Kunde König, nur in dieser Sache scheinbar ohne Amt und Würden !). Als Trostpflaster erhalte ich eine neue Nummer, wo mir angeblich weitergeholfen werden kann.

Bis heute ist es mir nur gelungen herauszubekommen, daß sich hinter dieser Nummer ein Techniker verbirgt, der solche Kisten wohl schon einmal repariert hat. Nur konnte ich ihn bis heute nicht an den Apparat bekommen.

Solchermaßen gesteßt, wurde das Problem mittlerweile recht elegant gelöst: bei HW-Elektronik in Hamburg habe ich mir alle Teile für einen 386SX gekauft und der XT zur Verschrottung freigeben.

Tschüs

*Helmut*



\* Keine Glosse, sondern Wirklichkeit... \*

Es wäre übertrieben, nachfolgenden Einakter als "Drama" zu bezeichnen. Aber ein Lustspiel ist es eigentlich auch nicht. Höchstens für den Außenstehenden. Für den betroffenen Akteur hätte es eine Tragödie sein können; aber der ist solchen "Kummer" gewohnt. Nennen wir es : einen traurigen (nicht tragischen) Sketch...

Ort der Handlung: Mannheim. Ein bekannter Supermarkt für elektronische Erzeugnisse: Nicht nur Computer, sondern auch Videogeräte, Fernseher, Radios, Kameras und und... Also lauter so'n technischer Kleinkram.

Zeit der Handlung: An der Schwelle des dritten Jahrtausend. Also weder in der Vergangenheit noch in der Zukunft. Nüchterne, unbarmherzige Gegenwart. Genauer gesagt: Mittagsstunde.

Bühnenbild: Riesenraum. Halb gedämpftes, weißliches Licht (kein Tageslicht, trotz Mittagsstunde). Viele Warengondeln, auf denen sich hochinteressante Technik im Kleinen türmt. Dazwischen Leute - Leute - teils mit suchenden Blicken, teils mit ernstem Blicke prüfend, teils gedankenvoll, teils -leer. Durchschnittsalter zwischen 10 bis 40, dazwischen Kinder (unter 10) und Greise (über 40). Überwiegend männlichen Geschlechts - ausgenommen bei den Fernsehern. Akustisch überwiegt Stille, gelegentlich durchdrungen von plötzlichen Schreien aus einer Ecke. Es ist die Ecke der CD-Player. Die Stimme hat KJ schon öfter (aus Versehen) gehört; nur kann er nie entscheiden, ob es Bryan Adams oder Adam Bryans ist. (Der Leser kennt sich da besser aus.)

Personen der Handlung:

1. Der Fachverkäufer in der Drucker-Abteilung (hier kurz "FV", sprich "EFF-AU!", genannt) ;
2. Der Kunde (hier kurz K.-J., sprich "KAJOT", genannt).

KAJOT, seit einem Jahrzehnt fast stolzer Besitzer eines EPSON RX80 F/T, also "nur Neunnadeldrucker", interessiert sich für einen möglichst EPSON-, auf jeden Fall aber IBM-kompatiblen Drucker mit 15 Nadeln mehr. (Wer mehr hat, is<s> mehr!)

Vorhang auf!

KJ: Guten Tag! Gestatten, ich möchte einen 24-Nadeldrucker!

FV: Sehr wohl, mein Herr! Wir haben da jede Menge!

(Was schon schlechter ist. Anm. der Regie.)

FV (zeigt auf einen PANASONIC): Was halten Sie z.B. von diesem hier? Der ist im Sonderangebot!

KJ: Was soll ich davon halten? Bitte, halten Sie mal meine Tasche.

(KJ kramt darin herum und holt schließlich einen Fragebogen heraus, den er sich zuhause zusammengestellt hat.)

KJ: Vielleicht halte ich viel davon, wenn er auch Proportionalschrift kann!

FV: ? ? ?

KJ: Ich meine: "Propro- äh - Popo- na, zum Gugguck, Sie machen mich ganz unsicher mit Ihrer Unsicherheit! Ich meine:

P r o p o r t i o n a l - - - und dann kommt -schrift!

FV: ? ? ? ?

KJ: Ja also, das ist so: man denkt, jeder Buchstabe sei gleich breit.

Is aba nich. I is dünner als O -

FV (unterbricht mit leuchtenden Augen): Aja, nahlar: I heißt IN- und O heißt OUTPUT. Jaja, einen Output hat er. Sehen Sie mal ...

(jetzt läßt FV doch tatsächlich eine ganze Seite ausdrucken, nachdem er elegant umständlich einen Bogen DIN A4 eingefädelt hat - was für KJ schon mal sehr wichtig ist, weil das beim EPSON glatter und somit schneller geht)

FV: Ja, was meinten Sie doch?

KJ: 'Proportionalschrift ist, wenn...' - - ja wenn jeder Buchstabe nur soviel Platz - (ach Gott, ich laß es lieber; hat wohl keinen Zweck) Sehen Sie hier im Ausdruck den Unterschied zwischen der Breite des i und des w ?

FV: ??? ???

KJ: Na schön. Dann sagen Sie mir bitte: Drucht der hier auch kursiv?

FV: ??

KJ (ich glaube, ich muß lauter sprechen) Ich meine: Schreibt der auch Italics?

FV: ? ? ?

KJ: Wissen Sie, das ist Schrägschrift.

FV: O ja, das tut er! Aber es ist doch besser, wenn der Drucker gerade steht. Eine kleine Wasserwaage bekommen Sie auch in unserem Hause.

KJ: Dankeschön! Die habe ich zu Hause. Ich stelle ihn also "gerade". ("Vielleicht läuft sonst die Tinte aus..." denkt KaJot)

FV: ?

KJ: Nun gut. Der Drucker scheint interessant. Besonders vom Preis her. Was kostet er doch nochmal?

FV (beflissen): hmhmhm Emmchen. Deutsche, versteht sich.

KJ: Sooo? In Ihrem Prospekt steht aber nur hmhm Emmchen. (Deutsche, versteht sich.)

FJ: Na ja, wenn Sie ihn kaufen, bekommen Sie ihn auch für hmhm Emmchen.

KJ: Ach so? Und was muß ich bezahlen, wenn ich ihn nicht kaufe??

FV: ??

P A U S E

(Beide erholen sich. KaJot guckt in den Prospekt, der neben dem Drucker liegt. Ihm gehen einige Lichter auf. \* \* \* Nach gebührender Warteschleife:)

KJ: Nun eine letzte, nein, lieber eine allerletzte Frage:

Im Prospekt steht nicht, ob der Drucker auch einen Download-Speicher besitzt. Das wissen Sie doch bestimmt!?

O Gott - Sie sind ja plötzlich so bleich! Ist Ihnen schlecht? Soll ich einen Kollegen rufen?

(Hier zieht der FV - sprich "Fachverkäufer" - kurz vor dem Zusammenbruch plötzlich ein kleines Ding aus der Rocktasche - KJ springt blitzschnell zur Seite, denn er kennt die rachsüchtige Unberechenbarkeit "technischer Kaufleute" und will Leben und Gesundheit noch etwas für seine Familie erhalten - - da spricht FV in das Ding leise hinein:

FV: Otto! Otto! Bitte zu Nummer 7a!

FV Otto kommt nach einigen bangen Minuten und wiederholter drahtloser Aufforderung.

KJ erblaßt vor soviel funktechnischer Sach- und Fachkenntnis und flüstert nur noch:

KJ: Guten Tag Herr Otto! Bitte geben Sie mir diesen PANASONIC. Aber nicht diesen, sondern Original-verpackt, bitte!

Otto: Zahlen Sie bar oder mit Scheck? Oder lieber auf Raten?

KJ: Ja was kostet der denn überhaupt?

Otto: Mein Herr, das ist ein Sonderangebot. Der kostet heute nur DM hm!

KJ: Soso; vorhin hätte ich fast hmhm bezahlt!

Also gut. Packen Sie ein. Inzwischen gehe ich zahlen.

Otto: ?

KJ: Jaja - ratlos - - ich meinte: ratenlos...in "DÄHMARK"!

Und vergessen Sie die Gebrauchsanleitung nicht, bitte!

Otto geht ab. -

Gekauft. (Heißt soviel wie: " E N D E ")

Otto heißt Otto. Das versteh' ich. Weil man's auch von hinten lesen kann. Was FV heißt, weiß ich aber nicht. Vielleicht: "Fröhlicher Vunker" ? ?

PS: Die Gebrauchsanleitung war wirklich drin! Ich bin sooo froh... Denn:

Ein Fachverkäufer ist versiert,

drum kaufe blind, was er serviert!

\*\*\*\*\*

*JOF*

Hier ist natürlich CP/M als Betriebssystem gemeint und auch das Thema ist nicht gerade neu. Im Jahr 1985/86 veröffentlichte Ted Silveira in "Computer Currents, 2550 9th Street Berkley, CA" eine Serie von Artikeln. Diese Artikel habe ich für unsere Verhältnisse überarbeitet, beziehungsweise als Vorbild genommen. Die originalen Texte sind bei mir zu erhalten, falls jemand daran interessiert sein sollte. Eigentlich gab das Lesen der Texte von Ted Silveira den Anstoß für die mich. Meist nimmt man sein eigenes Wissen und die persönliche Erfahrung als selbstverständlich hin vergisst aber, daß es oft Neueinsteiger gibt und viele einfach nicht so gut mit der Fremdsprache Englisch zurechtkommen. Da ich auch eine Menge an PD-Software aus den USA erhalten habe und vor obigem Problem stand habe ich viele PD-Programme getestet bzw. jetzt im Gebrauch. Natürlich ist der Einsatz meiner 68 MB Festplatte hier von Vorteil und vereinfacht die Handhabung sehr.

Die Texte von Ted Silveira sind copyrighted und ich habe hier nicht die Texte übersetzt, diese wären dann doch nicht gerade zeitgerecht, sondern angeregt dadurch einige Artikel verfaßt, welche in den nächsten INFO's erscheinen.

Meine Artikel orientieren sich aber an der Vorlage und Beschreibungen von Programmen sind natürlich ähnlich.

Vor zwei Jahren kaufte ich ein Modem und hatte im Nahbereich eine Mailbox mit CP/M PD-Software. Natürlich stiegen meine Gebühren um das 5fache und oft war das empfangene Programm nicht die Zeit wert.

Die Qualitäten der Programme sind sehr verschieden und wenn ihr etwas Einblick in die MSDOS - PCSIG habt (10000 PD-Programme), könnt ihr euch den Aufwand, ein geeignetes Programm für den eigenen Bedarf zu finden, vorstellen.

Aber der Mensch (ich auch) ist natürlich neugierig und so verbrachte ich oft Stunden vor dem Monitor und einigen Kisten mit PD - Software.

#### Was ist Public Domain Software ?

Public Domain heißt auf deutsch öffentlicher Bereich, aber eine

wörtliche Übersetzung ist hier wohl nicht ganz korrekt.

Einige Leute verbringen ihre Zeit am liebsten mit dem Schreiben von Programmen. Nicht weil sie dadurch Millionär werden wollen, sondern aus Spaß am konstruktiven Denken, weil sie in die Technik vernarrt sind oder eben jeder so seine kleinen Hobbies pflegt.

Wenn nun die Arbeit der "Programmierer" beendet ist werden die Programme der "Öffentlichkeit" anvertraut. Das heißt, jeder kann die Programme kostenlos bekommen und keine Lizenzbestimmungen müssen unterzeichnet werden. Man kann die Programme kopieren, ändern, weitergeben, also machen was man möchte. Damit aber keiner sich mit fremden Federn schmückt haben einige Autoren dies nur unter Beibehalt ihres Copyrightes gestattet, wie zum Beispiel bei MEX oder den Programmen zum ZCPR. Im allgemeinen sollte der Urheber der Programme sowieso immer erwähnt werden.

Ich bleibe mal bei dem Kürzel PD für Public Domain Programme. Viele PD-Programme sind nützliche Hilfen und ich möchte einige nicht missen. Einige Programme sind nur etwas für Programmierer, aber die meisten sind für den normalen Anwender.

XDIR zum Beispiel listet das Direktorie alphabetisch sortiert, zeigt die Dateilängen und den freien Platz auf der Diskette an.

BAD oder FINDBAD testet die Diskette und/oder Festplatte auf Fehler, zeigt die defekten Sektoren an und erstellt eine Datei welche diese Sektoren belegt, damit nicht darauf zugegriffen wird. NSWEEP, eines meiner Favoriten, kopiert, listet, druckt und vereinfacht das Kopieren enorm.

Wenn auch die Hilfsprogramme den überwiegenden Teil der PD ausmachen, so gibt es Programme für fast alle Bereiche. Für Datenübertragung, Anwenderprogramme für Datenbanken, Textprogramme, Erweiterungen zum CP/M (mein Lieblingsthema) und und und...

Ich erspare mir hier ein paar Seiten und möchte eine oft gestellte Frage aufgreifen.

#### Wie gut ist Public Domain Software ?

Sicher wollen viele das wissen und PD-Programme können sehr gut sein, aber die meisten sind eben doch nur 'Eigenbau', was aber

bitte nicht abwertend aufgefaßt werden soll.

Die meisten Programme sind älter und CP/M oder auch NEWDOS sind halt in den Anfängen der Computertechnik entstanden. Aber wer kann erlauben, mit welcher Freude ich meinen Genie einschalte oder mit dem MORROW die diversen DFÜ-Programme ausprobiere ?

Letztendlich haben auch so manche PD-Programme die Richtung der Softwareentwicklung bestimmt (nicht nur unter CP/M) .

Hierzu gehören zum Beispiel MDM7 , MEX, ZMP, im DFÜ-Bereich, NSWEEP oder NULU bei den Utilities und so großartige Sachen wie Richard Conn's ZCPR , oder die Autoinstallversionen NZCOM und Z3PLUS um nur einige zu nennen.

Alle PD-Programme haben aber eins gemeinsam, wenn es nicht klappt und die Arbeit der letzten Stunden verloren sind ist man alleine und fast immer auf sich selber angewiesen. Man kann Freunde fragen oder in Mailboxen um Hilfe bitten, aber es gibt keine Unterstützung und keine Hotline wie oft bei kommerziellen Produkten. Leider ist der Konkurrenzkampf heute so groß, daß der Endanwender oft als Beta-Tester herhalten muß und dann ist es mit der Unterstützung durch den Softwareentwickler eh nicht weit her.

#### Wie bekommt man PUBLIC DOMAIN SOFTWARE ?

Einer der einfachsten und angenehmsten Arten, PD-Software zu erhalten ist, dies über Mailboxen zu machen. Leider haben viele von Euch kein Modem und selten gibt es Mailboxen, welche alte Software , wie ich die für Newdos und CP/M hier bezeichnen möchte, auf der Festplatte haben.

Man darf hier nicht vergessen, daß alte Rechner unser Hobby sind, und diese sind halt nicht mehr Up-To-Date. Also ,am besten erhaltet Ihr die gewünschte PD-Software über einen User-Club, durch Anfrage bei unseren Soft-Archivaren, oder von Gleichgesinnten.

Es gibt durchaus einige Clubs ähnlich wie unser , deren Adressen werde ich aber erst im nächsten Info bringen.

Ich selber installiere gerade eine ZNODE auf einem ALTOS 580 Rechner. Diese ZNODE wird als DEMO-Rechner bei mir ab Mitte Februar laufen. Der Rechner wurde mir von Herrn Helmut Jungkuntz freundlicherweise zur Verfügung gestellt und so können sich Interessenten am Z-System selber ein Bild machen.

Danke Helmut!

ZNODE wird die Mailbox genannt , weil die Mailboxsoftware auf einem Z-80 Rechner nur mit den Utilities der CP/M Erweiterung NZCON oder ZCPR läuft und es halt meist um den Z80 geht.

Hierzu aber später mehr.

An PD-Software gibt es die Software der ursprünglichen USER-Groups , wie CP/M-UG, SIGM, PICONET und die deutsche Osborne User-Group, um nur einige zu nennen. Diese Gruppen haben sich im allgemeinen aufgelöst , oder sind zu moderneren Betriebssystemen gewechselt. Hierdurch entsteht oft das Problem, einen bestehenden alten Rechner in Gang zu bekommen, wenn keiner mehr weiss wie. Ich könnte euch da einige längere Geschichten von meinen Sammelstücken erzählen und konnte so manchem mit Hilfe der Clubkollegen zur Seite stehen.

Die Softwarelisten der User-Groups und neuer Software aus den USA sind bei mir , oder unseren Archivaren zu erhalten und es wird wohl kaum ein nicht lesbares Diskettenformat für uns geben.

In den nächsten Ausgaben des Club-Infos werde ich gute PD-Software vorstellen und zum Teil näher darauf eingehen. Da ich neben meinem MSDOS-Rechner fast nur mit CP/M arbeite sind die NEWDOS und GDOS-User hier leider etwas benachteiligt.

Deshalb !

Wer mit NEWDOS und/oder GDOS arbeitet möchte dem Club doch bitte seine Erfahrung mit der Soft- und Hardware mitteilen. Es gibt noch viele von uns, welche sich sehr aktiv mit diesen Rechnern befassen, unter anderem zu Steuerungszwecken oder als Wetterstation.

Genau ! IHR seid gemeint und nun trau EUCH mal zu schreiben.

Und was schreib ICH im nächsten INFO ?

Natürlich werde ich über DFÜ und die dazu nötigen Programme schreiben, vor allem da ich mich wegen der ZNODE gerade damit beschäftige.

Wie heißt es so schön

WER SCHREIBT - DER BLEIBT,

allerdings in einem anderen Zusammenhang, aber es könnte ja auch hier stimmen.

## Impressum

1. Vorsitzender            Fritz Chwolka  
Tel.: 0 24 64/89 20            Saarstraße 34  
   5173    Aldenhoven
2. Vorsitzender            Gerald Schröder  
Tel.: 0 41 05/26 02            Am Schützenplatz 14  
   2105    Seevetal 1
- Hardwarekoordinator      Andreas Magnus  
Tel.: 02 08/87 02 30            Bismarckstraße 29  
   4650    Gelsenkirchen
- Newdos-Diskotheke        Oliver Volz  
Tel.: 07 11/7 35 38 17        Waldenburgstraße 73  
   7000    Stuttgart 80
- CP/M-Diskotheke         Rüdiger Sörensen  
Tel.: 0 61 31/32 08 60        Thomas-Mann-Straße 3a  
   6500    Mainz 1
- Clubbücherei                Kurt Müller  
Tel.: 0 41 52/7 06 43        Sophie-Scholl-Ring 3b  
   2054    Geesthacht
- Redaktion                  Jens Neueder  
Tel.: 07 91/4 28 77            Rudolf-Then Straße 32  
   7178    Gschlachtenbretzingen
- Autoren                    Die Redaktion bedankt sich bei den im  
   INHALTSVERZEICHNIS genannten Autoren  
   für die Mitarbeit an der Club-INFO.

### Bankverbindung des CLUB 80

Postgirokonto Sonderkonto CLUB 80  
   Obermann H. 8870 Günzburg  
Konto Nr. 496 071-605 Postgiroamt Frankfurt BLZ 500 100 60

Eine Zensur oder Kontrolle der INFO-Beiträge erfolgt nicht.  
Die Redaktion.

# Schluß

Hallo Club 80'er,

wieder einmal haltet Ihr Euer neues Club-INFO in den Händen, und wieder einmal ist ein Jahreswechsel vorübergegangen. Den an dem Club-INFO mitwirkenden Kameraden möchte ich meinen Dank für die Mitarbeit aussprechen.

Gleichzeitig auch -im Nachhinein noch- meine allerbesten Wünsche für das jetzt begonnene Jahr an alle Club-Mitglieder. Hoffen wir, daß es im neuen Jahr mindestens so gut weitergeht wie im vergangenen Jahr.

Zum Abschluß dieses INFO's habe ich noch zwei Dinge, die mir am Herzen liegen:

Bitte sendet mir Euere Paper-Ware Artikel auf weißem Papier unter Verwendung eines frischen Farbbandes. Es ist für die Erstellung des INFO's wichtig, kontrastreiche Vorlagen zu haben.

Als Zweites, würde ich mich freuen, wenn sich eine BASIC-Anwender-Gruppe hier im Club finden würde. BASIC ist ja als Programmiersprache ein verbindendes Objekt zwischen all unseren Computern. Ich arbeite derzeit mit Quick-Basic 4.5 und PDS 7.1 als BASIC-Dialekte und bekenne mich als BASIC-Programmierer. Gibt es noch Gleichgesinnte?

Viel Spaß-  
bis zum nächsten mal

Jens

## Adressen-Liste CLUB80

Stand vom 12-02-1992

Seite 1

Nachname	Vorname	Straße	PLZ	Ort	Telefon	
					privat	Telefon ■ geschäftlich
Behrendt	Detlef	Schlosserbrite 1a	W- 8018	Grafing	08092 / 9173	■
Benner	Achim	Vorn Mühlberg 1	W- 5910	Kreuztal 3	02732 / 3780	■ (ab 18.00Uhr)
Berndt	Wolfgang	Friedberger Straße 92c	W- 6360	Friedberg 2	06031 / 2963	■
Berndt-Jochum	Ilse	Stachelsgut 24	W- 5060	Bergisch-Gladbach 1	02204 / 65254	■
Bernhardt	Helmut	Hafenstraße 7	W- 2305	Heikendorf	0431 / 241907	■ 0431 / 74047
Betz	Heinrich	St. Wolfgangstraße 13	W- 8551	Hausen	09191 / 31698	■ 09191 / 611108
Brans	Jörg	Tieloh 55	W- 2000	Hamburg 60	040 / 6906531	■
Braun	Harald	Postfach 8011	W- 2300	Kiel 17		■
Böckling	Ulrich	Jochaczstraße 61	W- 5410	Hör-Grenzhausen	0264 / 4861	■ 02631 / 895168
Chwolka	Fritz	Saarstraße 34	W- 5173	Aldenhoven	02464 / 8920	■
Dose	Volker	Dorfstraße 10	W- 2304	Brodersdorf	04343 / 1357	■
Drechsler	Oskar	Duckterather Busch 2	W- 5060	Bergisch Gladbach 2	02202 / 55282	■ 0228 / 128114
Gill	Thomas	Maria-Eich-Str. 34	W- 8000	München 60	089 / 8349527	■
Günther	Jens	Bannerscheid 7	W- 5231	Nettersen	02681 / 1553	■
Halgasch	Gert	Großschonauer Straße 26	O- 8805	Jonsdorf		■
Hartmann	Hans-Günther	Möwenstraße 9	W- 2876	Berne 2	04406 / 6911	■ 0421 / 2482419
Hebecker	Ulrich	Büsumerstraße 15	W- 7800	Stuttgart	0711 / 734800	■
Held	Manfred	Stirner Straße 22	W- 8035	Flintbek	09144 / 6563	■ 0911 / 2195245
Hermann	Klaus	Forchenstraße 8	W- 7401	Pliezhausen	07127 / 71945	■
Hürdler	Manfred	Niederhofer Straße 29	W- 8709	Rimper	09365 / 4235	■
Johnen	Willi	Hansemannstraße 1	W- 5160	Düren	02421 / 501305	■
Kauka	Dietmar	Straße des Friedens 37	A- 7201	Neukirchen	0037404/850112	■
Kemmer	Jürgen	Dorfberg 7	W- 8701	Sulzdorf	09334 / 1050	■
Kostya	Mary Jo	Balberstraße 68	CH- 8038	Zürich	00411 / 4828948	■ 00411 / 258170
Kuhn	Eckehard	Im Dorf 14	W- 7443	Frickenhausen	07022 / 45417	■
Littmann	Claus	Plockhorst, Zum Spring 15	W- 3155	Edemissen	05372 / 7796	■
Lorenz	Walter	Mahräckerstraße 9	W- 6000	Frankfurt /Main 50	069 / 531656	■
Magnus	Andreas	Bismarckstraße 29	W- 4650	Gelsenkirchen	0209 / 870230	■
Mahlert	Herbert	Hohenbudbergerstraße 112 A	W- 4100	Duisburg	02135 / 47217	■
Mand	Harald	K1. Flintbeker Straße 7	W- 2302	Flintbek /Kiel	04347 / 3629	■ 0431 / 3013580
Menk	Christian	Ollsener Straße 52	W- 2116	Hanstedt	04184 / 7825	■
Müller	Kurt	Sophie-Scholl-Ring 3b	W- 2054	Geesthacht	04152 / 70643	■ 04151 / 889137
Mösse	Franz	Schafferstraße 12	I- 39012	Meran		■
Neebe	Gerhard	Mürkische Straße 186	W- 4600	Dortmund 1	0231 / 416549	■
Neueder	Jens	Rudolf-Then-Straße 32	W- 7178	Gschlachtenbretzingen	0791 / 42877	■ 0791 / 44-4722
Neumann	Christof	Zeitblomstraße 22/2	W- 7910	Ulm		■
Nitschke	Stefan	Germanenstraße 5	W- 7519	Walzbachtal 1	07203 / 452	■
Obermann	Hartmut	Mozarting 23	W- 8870	Günzburg	08221 / 30248	■
Peters	Jürgen	Heukoppel 14	W- 2000	Hamburg 14	040 / 6412371	■
Reit	Hermann	Vechter Hof 40	W- 4500	Osnabrück	0541 / 16331	■
Retzlaff	Bernd	Kleiner Sand 98	W- 2082	Uetersen	04122 / 43551	■ 04103 / 605310
Riechmann	Michael	Letelner Heideweg 12	W- 4950	Minden	0571 / 36627	■
Rinto	Gerd	Rennbahnstraße 9	W- 2000	Hamburg 74	040 / 6552630	■
Ruschinski	Claus	Pommernstraße 21	W- 4370	Marl	02365 / 34646	■
Schilling	Werner	Ehndorfer Straße 340	W- 2350	Neumünster	04321 / 61116	■
Schimmer	Jörg	Castelling 55	W- 6369	Nidderau 1	06187 / 25503	■
Schmid	Alexander	St. Cajetan Straße 38/VII	W- 8000	München 80	089 / 495326	■
Schmitz	Rainer	Dornierstraße 17	W- 7320	Göppingen	07161 / 22549	■
Schober	Frank-Michael	Weberweg 2	O- 7590	Spremberg	0037574/ 4565	■
Schoberth	Uwe	Petrus-Waldus-Straße 14	W- 7136	Oetisheim	07041 / 7254	■ 0711 / 2011455
Schroers	Horst-Dieter	Breslauer Straße 9	W- 8016	Feldkirchen	089 / 9032615	■ 089 / 9032615
Schröder	Gerald	Am Schützenplatz 14	W- 2105	Seevetal 1	04105 / 2602	■
Schröder	Peter	Theodor-Fahr-Straße 32	W- 2000	Hamburg 62	040 / 5311582	■
Schröer	Egbert	Joachimstraße 18	W- 4270	Dorsten 1	02362 / 75840	■ 02365 / 495183
Schulte	Hartmut	Entenschnabel 8	W- 3162	Uetze	05173 / 1248	■
Schut	Andre	Sanderstraße 26	W- 1000	Berlin 44	030 / 6917861	■
Schwarz	Wolfgang	Schwedenring 6	W- 8850	Donauwörth	0906 / 3092	■
Seelmann-Eggebert	Jörg	Henri-Spaak-Straße 96	W- 5305	Alfter 4	0228 / 643853	■
Soerensen	Svend A.	Bogholder Allee 76A	DK- 2720	Vanløse		■
Stumpferl	Stefan	Hasenbergstraße 57	W- 8000	München 45	089 / 3144001	■
Sörensen	Rüdiger	Thomas-Mann-Straße 3a	W- 6500	Mainz 1	06131 / 320860	■ 06131 / 395268
Tornow	Wilhelm	Görlitzer Straße 16	W- 2190	Cuxhaven 13	04723 / 1355	■ 04723 / 3063
Vollkmer	Richard	Am Spörkel 69	W- 4600	Dortmund 50		■
Volz	Oliver	Waldburgstraße 73	W- 7000	Stuttgart 80	0711 / 7353817	■
Werner	Heiko	Reichenberger Straße 5	O- 8032	Dresden		■
Wittkamp	Heinz	Hindenburgstraße 37	W- 5630	Remscheid	02191 / 75132	■
Wulf	Hans-Otto	Im Brahmkamp 38	W- 4250	Botrop	02041 / 688972	■