

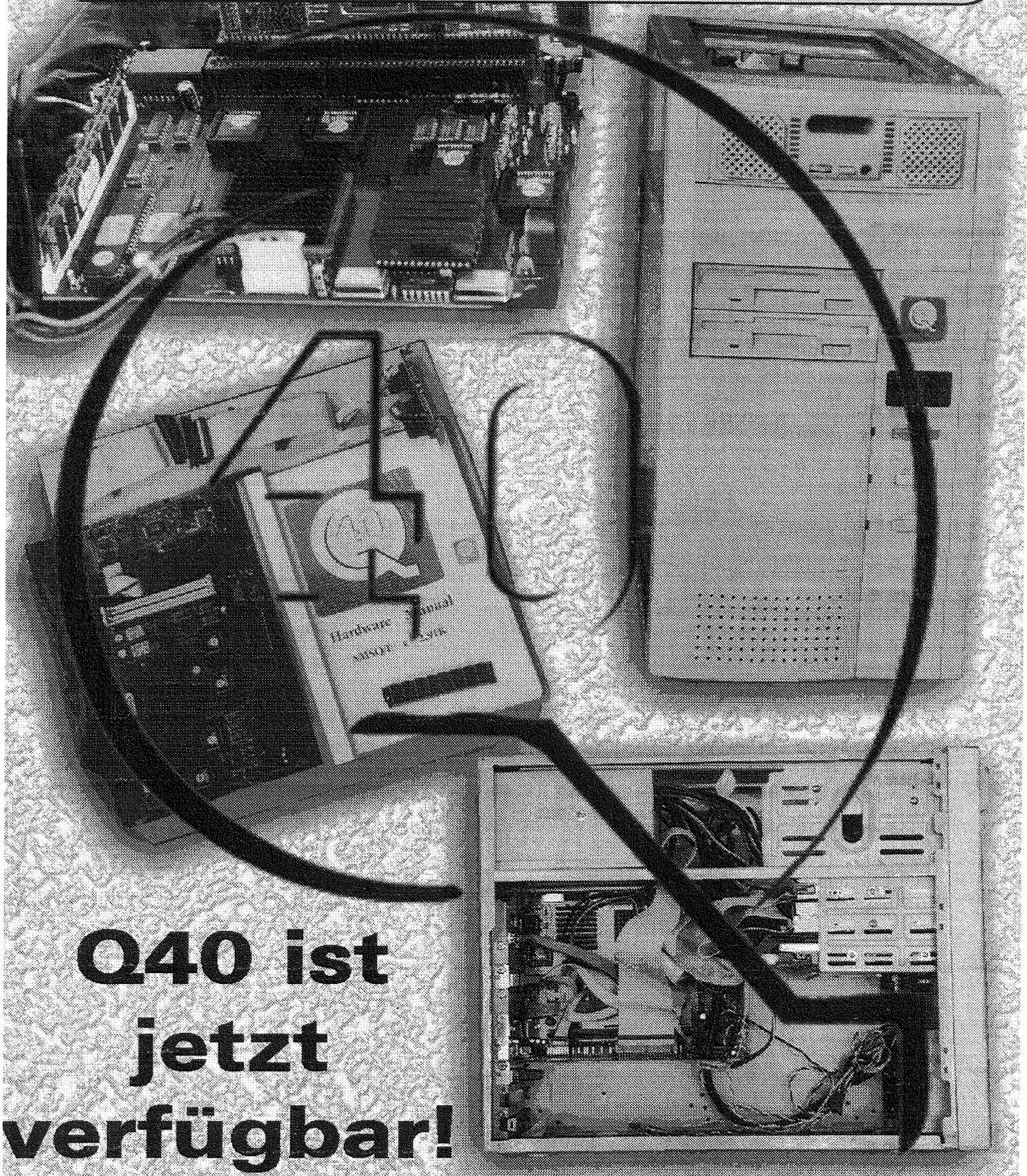
# QL Today

DEUTSCH

Jahrgang 4  
Ausgabe 1  
Mai/Juni  
1999

ISSN 1432-5446

Das Magazin über QL, QDOS,  
Sinclair Computer, SMSQ...



**Q40 ist  
jetzt  
verfügbar!**

# Inhalt

- 2 Kleinanzeigen
- 2 Impressum
- 3 Editorial
- 4 Zusammenfassung QL Today Volume 3,  
Issue 6, März/April 1999 *Wolfgang Uhlig*
- 5 Mein BOOT-Programm *Wolfgang Lenerz*
- 9 Werbung Fa. Vogel
- 10 Der QemuLator in der Praxis  
*Gerhard Plavec*
- 13 Werbung Fa. JMS
- 14 Wieder einmal BASIC *Helmut Boll*
- 16 Darf ich mal kurz unterbrechen...?  
*Jochen Merz*
- 17 Die Cover-Disk
- RS QL-Treffen

## Kleinanzeigen

Da QL Today eine der wichtigsten Quellen für QL-Neuigkeiten ist, werben auch die meisten QL-Händler hierin. Nun stellt sich natürlich die Frage, warum nur QL-Händler inserieren dürfen, warum nicht auch QL-User, die ihre eigenen Programme, Hardware oder Entwicklungen verkaufen möchten.

Anfangs konnte man selbstgeschriebene Software von Clubs vertreiben lassen, beispielsweise dem deutschen QL User Club. Aber warum soll man es denn nicht selbst vermarkten können? Es gibt hier keinen Unterschied zwischen privaten und kommerziellen Anzeigen, nur mit dem QL und Drumherum sollte es schon etwas zu tun haben.

Bis zu 50 Worte im englischen oder deutschen Teil kosten DM 5,- (oder 3 Internationale Antwortscheine), bis zu 100 Worte kosten DM 10,- (oder 6 Internationale Antwortscheine). Soll die Anzeige im deutschen und englischen Teil erscheinen, ist der Preis zu verdoppeln.

### Zu verkaufen

Mehrere Mega ST's, 1040, 520 von Atari und engl. oder dt. QL's, sowie diverses Zubehör wie ELKA CR 99 Midi Disk Recorder, Monitore etc.. Preis jeweils VS.

Manfred Waeser, Erfstadt,  
Tel. 02235/74495  
Email [waeser@t-online.de](mailto:waeser@t-online.de)

## QL Today DEUTSCH

ISSN 1432-5446

Herausgeber:

Jochen Merz Software    Tel.    +49 203 502011  
Im stillen Winkel 12    Fax    +49 203 502012  
47169 Duisburg    Box1    +49 203 502013  
Deutschland    Box2    +49 203 502014  
Email: [JMerz@j-m-s.com](mailto:JMerz@j-m-s.com)

**QL Today** erscheint alle zwei Monate, Erscheinungsdatum der ersten Ausgabe ist der 15. Mai. Das Abo beginnt mit der aktuellen Ausgabe zum Zeitpunkt der Bestellung. Preisinformationen und Antragsformulare sind bei Jochen Merz Software erhältlich.

Ihre Kommentare, Vorschläge und Artikel sind herzlich willkommen. SIE machen **QL Today** möglich. Wir verbessern das Magazin wo immer möglich, um Ihren Vorstellungen gerecht zu werden. Artikel sollten auf 3,5" Diskette (DD oder HD) eingeschickt werden. Das Format sollte ASCII, Quill oder Text87 (Druckertreiber angeben!) sein. Bilder sollten im \_SCR-Format geschickt werden, GIF und TIF ist auch möglich. BITTE senden Sie auch einen Ausdruck der Bilder. Wenn ein Bild an einer bestimmten Stelle plaziert werden soll, geben Sie es bitte auch an. Natürlich können auch alle Bilder/Artikel auf elektronischem Weg übersandt werden, also in die Box oder per E-Mail.

### Redaktionsschluß für Artikel und Werbung:

Ausgabe 1:	30. April
Ausgabe 2:	30. Juni
Ausgabe 3:	30. August
Ausgabe 4:	30. Oktober
Ausgabe 5:	30. Dezember
Ausgabe 6:	30. Februar

**QL Today** behält sich vor, eingeschicktes Material nicht zu veröffentlichen. **QL Today** ist unter keinen Umständen für die Richtigkeit der abgedruckten Artikel und Programmen haftbar, ebenso nicht für aus fehlerhaftem Material hervorgerufene Datenverluste, Unbenutzbarkeit oder ähnliche Probleme, die aus Artikeln in **QL Today** herrühren könnten. Die Meinung in diesem Magazin entspricht der des jeweiligen Autors und nicht notwendigerweise der des Herausgebers.

Dieses Magazin unterliegt dem Copyright und jegliches hierin veröffentlichte Material darf nicht ohne schriftliche Erlaubnis von **QL Today** reproduziert, übersetzt oder sonstwie verbreitet werden. Allen Copyrights und Trademarks wird hiermit Rechnung getragen.

Liebe Leser,

vielen Dank für Eure Treue - nun sind wir schon im vierten Jahr!

Dieses Mal schreibt sich das Editorial fast von alleine, so viel Neues ist passiert:

Da wäre zuallererst der Q40 - nun fertig, tummelt er sich schon bei einigen Kunden. Wir hatten ja in vergangenen Ausgaben etwas über seine Fähigkeiten geschrieben. SMSQ/E für ihn gibt es auch schon, es klappt fast alles bis auf die erweiterten Bildschirm-Auflösungen. Es scheint, daß der Bildschirm Tony besondere Kopfzerbrechen bereitet, jetzt wartet noch eine Hardware mehr auf seine neuen Bildschirmtreiber. Wer einen Q40 besitzt: Bitte schreibt uns einen Erfahrungsbericht!

Bei JMS gibt es einen neuen QD (hier wurden nur kleine Kundenwünsche realisiert, z.B. ist die Farbe der kleinen Hilfe-Popups konfigurierbar) - heißt nach wie vor QD98 und Update ist wie immer kostenlos.

Beim neuen FiFi wurden einige Fehler beseitigt - Update auch hier kostenlos. Ganz neues Highlight: QSpread99 - ich will hier im Editorial aber nicht zu viel Eigenwerbung betreiben, Details gibt es in der JMS-Werbung.

Updates sind wie immer in der JMS-Box verfügbar. Hierzu noch ein paar Kommentare: Die Ferngesprächs-Kosten während der Nacht sind ja jetzt über andere Telefon-Anbieter auf Ortsgesprächs-Niveau der Telekom gesunken. Ich will hier keine Werbung für bestimmte Firmen machen, doch sind 4 bis 5 Pfg/Minute kein Problem mehr. Einfach mal informieren! Weiterer Vorteil: Wer für das Modem störende Gebühren-Impulse von der Telekom erhält, der bekommt sie bei anderen Anbietern nicht mitgeschickt.

Eigentlich sollte jetzt die Box gestürmt werden, es gibt nämlich nicht nur Updates sondern auch eine Menge interessanter kostenloser Software. Schaut doch einfach mal hinein!

Mit dieser Ausgabe gibt es ja die erste Cover-Disk für die deutsche Ausgabe. Wir haben alle möglichen Listings aus den vergangenen Ausgaben drauf gepackt, zudem noch einige schöne Spiele und das absolute Highlight zur Zeit: Paragraph, die neue Textverarbeitung! Wer ProWesS besitzt, kann sie kostenlos ausprobieren.

Ich habe sie selber mal gestartet (keine Installation notwendig, nur ProWesS wird vorausgesetzt). Wenn mehr Infos zu ProWesS gewünscht werden, dann schreibt bitte! Wie gesagt, ich habe ProWesS ausprobiert und war sehr positiv überrascht. Skalierbarer Text, farbige Ausdrücke auf meinem EPSON Stylus Color, Einbinden von Grafik und Bildern (QL-Format und sogar GIF!) ... und alles funktioniert! Sogar Tabellen lassen sich automatisch anlegen. Ich hatte bei dem ganzen Herumprobieren nur einen Absturz, nämlich als ich einen leeren Block formatieren wollte. Der Autor arbeitet am Programm, Fehler werden sofort korrigiert. Man kann (und sollte) sich registrieren lassen - es kostet nicht viel. Die Version für registrierte Nutzer kann noch deutlich mehr.

Probiert es aus, es ist intuitiv ohne Anleitung zu nutzen (Anleitung liegt zudem noch als HTML-Text bei, ist mit dem ProWesS-Reader zu lesen und zu drucken). Da Paragraph auf ProWesS basiert, kann es alle Vorzüge nutzen: viele verfügbare Druckertreiber, die nicht umständlich angepaßt werden müssen, Farbausdruck, Treiber erweiterbar... wirklich sehr empfehlenswert. Ich muß gestehen, ich war bislang nichtallzu sehr von ProWesS angetan, doch in der aktuellen Version ist es deutlich schneller, auch kleinere Schriften sehen besser aus, die Installation ist sehr einfach, und wir haben eine Top-Anwendung, die alleine schon den Kauf von ProWesS rechtfertigt.

Bis in zwei Monaten *Jochen Merz*

# Zusammenfassung QL Today Volume 3, Issue 6, Jan/Feb 1999

Wolfgang Uhlig

## Neuigkeiten

### Turbo Toolkit V3c27

Eine noch neuere Version des TurboToolkits, das die b-Version, über die erst in der letzten QL-Today berichtet wurde, ersetzt. Es sollte auf allen einschlägigen QL-Adressen zu finden sein und sollte jetzt eigentlich auf allen Plattformen gut funktionieren. Es gibt auch ein Textfile dazu, in dem alle Funktionen des neuen Turbo Toolkit erklärt werden.

### PERFECTION

Eine "runderneuerte" Version von Perfection wird im Moment Beta-getestet und wird bald erscheinen. Perfection nutzt dann auch alle Bildschirmauflösungen und wird konfigurierbar sein.

### Überall einsetzbares Screen-Toolkit

Mark Knight hat ein Toolkit entwickelt, mit dem Programmierern eine sichere Möglichkeit an die Hand gegeben wird, mit verschiedenen Bildschirmauflösungen umzugehen, Fenstergrößen zu verändern, usw. Es wird freeware sein. Vom selben Autor wird es demnächst eine Sammlung von Fraktalprogrammen geben.

### C68 Support Seiten

Dave Walkers C68 Support-Seiten haben eine neue Internetadresse:

<http://www.itimpi.freeseve.co.uk>

Von dieser Adresse kann man auch George Gwilts software herunterladen.

### Superbasic Sourcebook von Tim Swenson

Es umfasst Qliberator, Toolkits und andere Programmierertools. Erhältlich auf der Seite:

[www.geocities.com/SiliconValley/Pines/5865](http://www.geocities.com/SiliconValley/Pines/5865)

### MicroEmacs Editor

Thierry Godefroy stellt auf seiner Webseite die PointerVersion des o.g. Editors zur Verfügung  
<http://www.imagnet.fr/~godefroy/english/download.html>

### Neue Software vom Anfänger-Club Italien

Ein Offline-Reader unter dem Pointer-Environment und der "Euro Converter", ein kleines Programm-

chen, mit dem man alle europäischen Währungen in Euro umrechnen kann und umgekehrt.

## Secrets of the QL

### Geheimnisse des QL

von Al Boehm. Der Autor berichtet über das neue SBasic/SuperBasic Reference Manual von Rich Mellor, Franz Hermann und Peter Jäger, das neulich herausgebracht wurde und aus zwei knapp vier Zentimeter dicken "Schinken" plus drei Disketten besteht. Hier bleiben keine Fragen mehr offen: Die außergewöhnlichsten Toolkits werden behandelt, sogar die Befehle für das QL Soundboard (hat das jemals jemand gesehen?) **[Ja, gab's früher mal bei ABC-Elektronik, ich hatte auch eins - Jochen]** sind enthalten. Auf den Disketten finden sich ein Index mit Querverweisen, viele Beispielprogramme, die oft so in eigene Programme übernommen werden können und das Schönste: Alle Toolkits, aus denen Befehle stammen, sind auf den drei Disketten vorhanden!

## GEE Graphics! (On the QL?) Part 9

Herb Schaaf beschäftigt sich in diesem neunten Teil seiner 'unendlichen Geschichte' immer noch mit dem überaus mysteriösen Phänomen eines Tetrahedrons und wie man so etwas in einem QL 'bauen' kann. Diesmal geht es sogar darum, ein Stereo-Paar von Tetrahedra abzubilden. Dieser für Kenner sicher sehr interessante Artikel wird dieses Mal NICHT vervollständigt durch ein Listing, denn das würde etwa 50 Seiten in Beschlag nehmen. Aber wahrscheinlich ist ein Programm auf der Cover-Disk, in dem man nach Herzenslust herumschwirren kann. **[Stimmt, ist auf der Disk]**

## catdoc

Ein Erfahrungsbericht von Timothy Swenson, Catdoc ist ein Unix-Programm von Victor Wagner und wurde von Jonathan Hudson für den QL aufbereitet. Es ist für alle die gedacht, die in die Verlegenheit kommen, ein Word-Dokument lesen zu müssen, ohne Word zu haben. Diese Aufgabe erfüllt es zufriedenstellend, auch wenn man den entstehenden ASCII-Text ab und zu noch etwas nacharbeiten muss.

## You and Your Programs - Just good Friends?

Du und deine Programme - einfach nur gute Freunde?

Geoff Wicks lässt sich zwei Seiten lang über seine Programme aus und wie toll sie doch "designed" sind unter dem Vorwand, etwas Sinnvolles über das Bildschirm-Design von Program-

men zu schreiben. Ziemlich uninteressant für meine Begriffe!

## Electronic Publishing on the QL

von Tim Swenson

Der Autor beschäftigt sich erst im allgemeinen mit dem Problem des Publizierens auf dem Computer und der Notwendigkeit dafür, um danach auf die Möglichkeiten des Publizierens von Text und Graphik speziell auf dem QL einzugehen. Behandelt werden die verschiedenen Formate wie Text-Dateien, Quill-Dokumente, DJ-Viewer, HTML und MicroEmacs HTML-Dateien und Viewer, ihre Vor- und Nachteile und Realisierbarkeit auf dem QL. Eine gründliche Analyse.

## The structure of \_ABA files

Die Struktur von \_aba-Dateien

von Christopher Cave. Eine Auflistung von Bits und Bytes einer \_aba-Datei für Leute, die ihre Abacus-Dateien z.B. in einem Textviewer darstellen wollen.

## QUILLRTF

Auf der Webseite <http://www.anit.es/pedro> gibt es ein Programm, das sich wie oben in der Überschrift nennt und genau das tut, nämlich Quill\_docs in RTF-Dateien umwandelt und dadurch viel Formatierungsarbeit erspart, wenn man seine

Quill-docs in der Windows Welt benutzen möchte. Timothy Swenson hats ausprobiert und ohne spanisch zu können, ist es ihm gelungen!

## Programmieren in Assembler - 4

von Norman Dunbar. Wieder ein Leckerbissen für Leute, die interessiert an diesem Thema sind. Sogar neun ganze Seiten lang!! Und immer mehr jenseits des Begriffsvermögens einiger anderer (so wie ich :() )

## Printer Control Codes - A dreaded Subject? - Teil 1

Druckerkommandos, ein beängstigendes Thema?

von Dilwyn Jones. Der Autor beschreibt, was Druckertreiber sind, wofür sie wichtig sind und wie man begreift, was da eigentlich passiert. Was sind eigentlich Drucker-Kontroll-Kommandos? Was ist ein ESCAPE-Character? Warum können nicht alle Drucker denselben Treiber haben? und, und, und... Des weiteren werden die Standard Kontroll-Code-Sammlungen vier verschiedener Druckerhersteller vorgestellt und ihre Unterschiede beleuchtet, nämlich Epson, IBM, Canon und Hewlett-Packard. Ich vermute, dass Dilwyn nächstes Mal ausführlicher auf die eigentliche Programmierung eingehen wird. **[Richtig vermutet, Wolfgang!]**

---

## Mein BOOT-Programm!

Wolfgang Lenerz

Es scheint einiges Interesse daran zu geben, zu sehen wie verschiedene Leute ihre Maschine booten, also hier ein paar Worte zu meinem eigenen Boot-Programm. Ich behaupte nicht, daß das irgendwie besonders lesenswert ist, aber vielleicht sind der eine oder andere Aspekt doch interessant.

Natürlich ist das Programm ganz auf mich zugeschnitten, also weiß ich nicht inwiefern es für Sie von Nutzen sein kann, aber hier ist es **[Sie finden es auf der Cover Disk - Editor]**. Bitte beachten Sie, daß main Boot-Programm über einen längeren Zeitraum hinweg entstand - ab und zu wird etwas hinzugefügt, wann immer es nötig ist. Das heißt, daß es da so einige Sachen gibt, die, wenn man sie neu schrieb, bestimmt besser geschrieben

sein könnten. Andere Zeilen sind vielleicht überflüssig. Es ist nur so, ich halte es mit dem Spruch: Wenn's funktioniert - bloß nicht ändern!

Sie werden sehen, daß ich den "dev" Treiber viel benutze, also den virtuellen Treiber. In einem der vorigen QL Todays schrieb jemand, er wüßte nicht wozu so etwas gut sein könnte. Ich persönlich könnte allerdings ohne das "dev" Gerät nicht leben!

(Nur einen kleinen Hinweis für

die, die dieses Gerät nicht nutzen oder kennen: dev ist ein virtuelles Gerät, der immer auf ein aktuelles Gerät zeigt. Zum Beispiel könnte man "dev1\_" so setzen, daß er auf "win1\_" weist. Wenn man dann so etwas wie "DIR dev1\_" eingibt, würde man in Wirklichkeit das Verzeichnis von win1\_ lesen. Man kann "dev1\_" auch auf etwas Komplizierteres setzen, so zum Beispiel "win1\_quill\_", so daß "DIR dev1\_" die Dateien im Unterverzeichnis "win1\_quill\_" anzeigen würde. Das ist für die alten Psionprogrammen ganz nützlich).

Um zu erklären warum mir dieses dev Gerät so nützlich ist, muß ich erst kurz auf meine Computerstation eingehen, oder vielmehr auf meine Computerstationen, da ich mehr als

eine habe(n muß). Aus verschiedenen Gründen, auf die hier nicht eingegangen werden muß (sonst würde ich Sie zu Tode langweilen), habe ich mehrere QL-haftige Maschinen: Zwei PC mit QXL, einen Schlepptop mit QPC, einen Atari Mega ST mit Emulator und, wirklich, noch so eine alte schwarze Box mit einer Goldcard (erste Version) und 2.88 MB Laufwerken. Ich hoffe, dies hört sich jetzt nicht nach angeben an, auf jeden Fall brauchte ich jede dieser Maschinen irgendwann mal: Eine QXL brauche ich im Büro, die andere zu Hause, meistens um da weiterarbeiten zu können. Der Laptop kommt natürlich mit auf Reisen. Der ST war lange mein Hauptcomputer bis die QXL erschien. Der QL wurde für Softwarebibliotheken gebraucht, da ich lange ein Quanta Sub-librarian war, und auch der Bibliothekar unseres französischen Clubs QLCF. Heute sind der ST und der QL halb in Pension gegangen, obwohl ich sie immer noch als Prüfsteine gebrauche. Ein Problem mit so vielen Maschinen ist das der Anpassung: Ich hätte gerne die Möglichkeit, von einer Maschine auf die andere zu springen ohne zu wissen (und ohne Wissen zu brauchen), mit welcher Maschine ich denn da jetzt gerade arbeite. Das war immer ein Problem, da die QXL unter SMSQ lief, der QPC unter SMSQ/E, der QL einen JS-ROM hatte, und der Atari wieder etwas anderes. Jedes dieser Betriebssysteme hatte so seine eigenen kleinen Macken. Deshalb war ich auch so froh als SMSQ/E entstand - denn jetzt habe ich EIN UND DASSELBE Betriebssystem für alle meine Maschinen - die haben jetzt alle die gleichen Möglichkeiten. Natürlich sind einige schneller als andere, aber das ist dann

nur Nebensache.

Nun gut, also habe ich die 'gleiche' Maschine überall, aber da haperte es immer noch an etwas: Tatsächlich haben ja alle Maschinen, außer dem QL, Festplatten aber die sind nicht alle "win1\_". Aus verschiedenen Gründen, liegt die "QXL.WIN"-Datei auf den PCs nicht auf dem Laufwerk C: (dann wäre sie ja WIN1\_ für SMSQ/E) sondern auf anderen Laufwerken (und, um die Sache ganz verrückt zu machen, liegt diese Datei auf dem Laufwerk D: auf einer Maschine, auf dem Laufwerk E: auf der anderen, und auf F: auf der dritten). Das heißt, daß während meine Dateien auf dem Atari alle in win1\_ liegen, auf dem PC sind sie in win2\_, win3\_ und win4\_. Auf dem QL liegen sie dann natürlich auf flp1\_. Wenn ich immer im Kopf behalten müßte, wo jetzt was bei welcher Maschine liegt, hätte ich ewig eine Migräne. Außerdem, wenn man ein Programm auf einer Maschine so konfiguriert, daß es seine Daten z.B. auf "win1\_aff\_data\_" sucht, müßte es man es auf der anderen so konfigurieren, daß es auf "win2\_aff\_data\_" sucht usw. Pfu!

Hier kommt natürlich das "dev" Gerät zu Hilfe: Ich brauche nur "dev1\_" so zu setzen, daß es auf die jeweilige SMSQ/E Festplatte zeigt (oder flp1\_ auf dem QL) und dann brauche ich immer nur auf dev1\_ zuzugreifen: Alle Programme können auch so konfiguriert werden, daß sie ihre Daten auf dev1\_ suchen (z.B. "dev1\_aff\_data\_"), und sie brauchen sich gar nicht darum zu kümmern, was dev1\_ genau ist oder wo es liegt.

Es bleibt also nur noch ein kleines Problem: Wie wird denn jetzt dev1\_ auf das jeweilige Laufwerk eingestellt? Ich bin natürlich viel zu faul dies jedes Mal zu tun, wenn ich eine

Maschine benutze (dann müßte ich ja im Kopf behalten welche Maschine welches Laufwerk benutzt, das ist ja undenkbar...). Also muß auch der QL wieder 'ran. Dies geschieht in der "find\_mydrive\$" Funktion, die in Zeilen 3150 ff. liegt. Diese Funktion wird früh in meinem Boot aufgerufen. Wie Sie sehen, setzt diese Funktion erst einmal den DATA\_USE default auf eine leere Zeichenkette (mehr dazu später). Dann versucht sie, in einer Schleife, ein Verzeichnis auf win1\_ bis win8\_ zu öffnen. Dies geschieht mit der FOP\_DIR Funktion. Wenn diese Funktion fehlerfrei zurückkommt, weiß ich, daß da eine gültige Festplatte liegt, die nötigerweise die erste QL Festplatte der Station ist. Also wird die Bezeichnung dieser Festplatte von der "find\_mydrive\$" Funktion zurückgegeben, und später wird das dev Gerät darauf gesetzt. Darum wurde auch DATA\_USE zu einer leeren Zeichenkette gesetzt: Die FOP\_DIR Funktion versucht ja zuerst das angegebene Verzeichnis zu öffnen (d.h. win1\_ in der ersten Schleifenrunde). Wenn das nicht klappt, dann versucht diese Funktion ein Verzeichnis zu öffnen, wobei der DATA\_USE Defaultwert vor das angegebene Verzeichnis gesetzt wird (d.h. wenn DATA\_USE auf flp1\_ gesetzt ist, würde die Funktion versuchen "flp1\_win1\_" zu öffnen). Wäre dann eine Diskette im Laufwerk 1, würde dies sogar erfolgreich sein (!) und das Ergebnis wäre dann falsch. Auf jeden Fall wird hier im Endeffekt "dev1\_" auf die korrekte Festplatte eingestellt, und sogar auf "flp1\_" im QL.

So, lassen wir uns jetzt mal das eigentlich Boot-Programm anschauen. Zuerst werden einige Variablen gesetzt, durch Aufrufen der "constants" Prozedur.

Diese Variablen werde ich später noch erklären. Die "ws" Prozedur wird dann aufgerufen, sie macht den ganzen Bildschirm erst mal schwarz und schreibt dann eine kleine angeberische Nachricht ziemlich in der Mitte des Bildschirms. Dafür werden die xlim und ylim Variablen genommen, welche in der "constants" Prozedur auf die X und Y Bildschirmgröße gesetzt wurden. Diese Größe ist natürlich von Maschine zu Maschine unterschiedlich. Der QL ist auf 512\*256 begrenzt. Auf den PCs benutzte ich 800\*600, und der Atari hat noch eine andere Größe. Die "ws" Prozedur funktioniert richtig, solange die Bildschirmgröße nicht kleiner als ein normaler QL Bildschirm ist.

Danach werden die Fenster etwas verändert und eine kleine Nachricht auf den Bildschirm gesetzt (Zeilen 150-190) - beachten Sie wie das Fenster#0 auf fast verschwindende Werte gesetzt wird. Das mache ich extra, da ich später eine ganze Menge Extensions lade. Wenn Fenster#0 auf diesen Wert gesetzt wird, können die Extension-Dateien nicht mehr ihre Initialisierungs-Meldungen über den ganzen Bildschirm schreiben, was ich immer als sehr unschön empfand. Ich weiß ja schließlich was ich lade und brauche nicht immer wieder darauf hingewiesen zu werden.

Danach rufe ich die "find\_mydrive\$" Funktion auf, um das Gerät zu finden, auf das dev1\_ gesetzt werden soll. Wenn dies aus irgendeinem Grund nicht klappt, beschwert sich das Boot-Programm und hört auch sofort auf (Zeilen 210-220).

Mit den Zeilen 240 bis 380 lade ich mir alle Extensions ein die ich brauche. Hier können einige Erklärungen auch wieder nüt-

lich sein. Diesen Teil habe ich zu meinem Boot-Programm gefügt als die QXL kam. Die Geschwindigkeit mit der eine QXL auf der PC Festplatte liest/schreibt ist furchtbar (langsam). Ich dachte, es wäre besser, alle Extension-Dateien in eine einzige große Datei zu legen, anstatt mehrere kleine Dateien zu laden (im ganzen, 35!). Also habe ich, mit Hilfe einer kleinen Basicprozedur die später erklärt wird, eine große Datei aus allen meinen Extensions gemacht, die werden einfach alle Ende an Ende gelegt. Das Resultat ist "dev1\_booty\_boot\_rext", deren aktuelle Länge in der Variablen 'tot\_len' enthalten ist. Diese wird in der "constants" Prozedur initialisiert und ist im Moment 303152 Bytes. Die Datei wird dann auf einen Schwung geladen, mit einem einzigen LBYTES Befehl. Die einzelnen Extensions werden dann in einer Schleife aufgerufen - darin werden der Name und die Länge der einzelnen Extensionsdateien gelesen, und so kann ein CALL auf den richtigen Platz innerhalb der großen Datei gemacht werden. Die Namen der Extension-Dateien werden ausgedruckt (das wurde nur für Debuggingzwecke eingeschoben, so daß, falls die Bootprozedur einmal stoppt, ich auch wissen kann, welche Extension das Problem darstellt - da ich selber einige von ihnen geschrieben habe, ist dies öfter auch ganz nützlich...). Die Namen der Extensions (Dateien) sind in DATA-Befehlen in Zeilen 2510 ff. aufgeführt.

Man muß nur darauf aufpassen, daß alle Dateilängen gerade Zahlen sind, sonst würde die nächste Extension an einer ungeraden Adresse aufgerufen, und das kann natürlich jede Art von Problemen erzeugen!

In Zeilen 380-430, nachdem ich Fenster#0 auf etwas Passen-

des gesetzt habe, werden auch die seriellen Schnittstellen auf für mich passende Werte gestellt. Diese Werte würden auf dem QL einige Probleme geben, aber den benutze ich für DFÜ sowieso nicht.

Die "use\_defaults" Prozedur, die in Zeile 440 aufgerufen wird, setzt die verschiedenen Geräte für dev1\_, PROG, DATA und DEST\_USE, usw. Da wird auch das SUB-Gerät gesetzt (noch ein virtuelles Gerät). Das ist nur noch wegen meiner Faulheit da, ich könnte auch das auf DEV1\_ umkonfigurieren.

Nachdem dann die Zeit/Datum ausgedruckt wird (über die "time" Prozedur - mehr davon später), wird eine spezielle Tastaturbelegung geladen (Zeilen 470-510). Die gibt mir ein CTRL-C auf F11 und ein CTRL-SPACE auf F12. Der "PUT\_KBD" Befehl kommt von der "clavier\_bin" Extension. "Clavier" gibt es auch auf einigen BBS und erlaubt es, die Tastatur eines PCs unter QXL und QPC zu ändern.

In den Zeilen 520-560 wird meine Buchhaltung geladen. Wie Sie sehen können, hängt es von der "do\_all%" Variablen ab, ob sie wirklich geladen wird. Wenn ich das REMark aus der Zeile 120 wegnehme, dann ist do\_all% gleich 0 in Maschinen mit weniger als 5 MB RAM, und manche Programme werden nicht geladen.

Direkt danach, in Zeilen 580 bis 680, lade ich einige andere Programme: den QPAC1 "Taschenrechner", und drei von mir entworfene Programme. Dies geschieht mittels einfacher Hotkey Befehle. Beachten Sie, daß ich die HOT\_RES, HOT\_REMV und HOT\_WAKE Kombination auch durch ein einziges HOT\_RES1 ersetzen könnte, aber durch Faulheit bleibt alles wie es ist.

Nachdem noch mein Büroverar-

beitungsprogramm geladen wird (Zeilen 690-7250), installiere ich einige QPAC2 Programme. Ich brauche normalerweise nicht alle auf einmal, aber sie werden dennoch installiert. Dies wird über zwei Prozeduren gemacht: "qpac2" (Zeilen 1470 - 1590) und "qp" (Zeilen 1610 bis 1680). Die "qpac2" Prozedur erstellt nur die Hotkeys und ruft dann die "qp" Prozedur auf, welche die Buttons in die Buttonleiste setzt, und auch den Hotkey Job startet. Ich setze nicht alle QPAC2-Programme in die Buttonleiste, da meine "professionellen" Programme auch die Buttonleiste benutzen, und ich möchte nicht allzu viele Knöpfe auf meinem Bildschirm herumliegen haben - ich finde, die nehmen mir zuviel Platz weg.

Danach lade ich dann einige Programme zum assemblieren und kompilieren von Assemblerprogrammen. Ich lade sie über HOT\_RES ein, aber dann schmeisse ich den Hotkey wieder weg: Der einzige Zweck hier ist, die Programme als ausführbare Things zu laden, wo sie dann immer im Speicher bleiben. Sie werden dann direkt aus dem Speicher benutzt, wann immer nötig (Zeilen 770-860).

Ich lade auch mein Rechnungsprogramm (Zeilen 870-920) und danach stelle ich einige oft benutzte Hotkeys ein, die meisten um meinen Namen ans Ende von Briefen zu setzen (Zeilen 930-990). Dann werden Hotkeys für QD eingestellt (Zeilen 1000 bis 1020). QD selbst ist eine von den Extensions die vorher geladen wurden. Ich erstelle zwei Hotkeys für QD, einen um existierende QDs zu picken/wecken (Zeile 1010), den anderen um ein neues QD auszuführen (Zeile 1020).

In Zeile 1030 lade ich ein

kleines Programm namens "blank". Wenn dies über den diesbezüglichen Hotkey ausgeführt wird ('&'), wird der ganzen Bildschirm geschwärzt, und es wird gewartet, bis ein Kennwort eingegeben wird. Der eindeutige Zweck ist, zu verhindern, daß jemand lesen kann was auf dem Bildschirm steht. Ich habe halt einen kleinen Sicherheitsfimmel... Durch dieses kleine Programm wird natürlich auch eine Programmumschaltung mittels CTRL-C verhindert!

Als nächster Schritt kann, wahlweise Xchange geladen werden oder nicht (hängt wieder von "do\_all%" ab). Wird es geladen, erstelle ich auch einen Hotkey, der dann nachher in die Tastaturschlange eingefügt wird. Dadurch führt Xchange ein kleines TSL Programm aus, das wiederum einige Voreinstellungen in Xchange vornimmt, automatisch Quill aufruft und ein bestimmtes Dokument lädt, welches, originalerweise, "default\_doc" genannt wird. So ist Quill dann sofort bereit und genauso eingestellt wie es mir gefällt.

Die nächsten paar Zeilen (1160-1190) erstellen wieder einige Hotkeys, wieder für Briefe, und einige spezielle Druckerpuffer. In Zeile 1200 wird erneut eine Sicherheitsanlage erstellt, die dem oben erwähnten 'blank' Programm sehr ähnlich ist. Dadurch wird ein Programm gestartet, welches im Hintergrund läuft. Es wird 'secure' genannt und, nach einer bestimmten Zeit ohne jegliche Aktivität (hier drei Minuten) wird der Bildschirm auch geschwärzt und es wird auf ein Kennwort gewartet. Mit anderen Worten, wenn ich drei Minuten lang nichts an der Tastatur eingebe, oder die Maus nicht bewege, wird der Bildschirm schwarz. Noch so

eine Sicherheitsfimmel-Einstellung, aber ganz nützlich, wenn man das Büro für einige Minuten verlässt.

Zeilen 1220 bis 1340 werden wieder nicht immer ausgeführt. Wenn sie ausgeführt werden, wartet man zuerst in einer Schleife, daß alle meine Büroprogramme sich initialisiert haben und als Button erschienen sind. Dies wird mit der "BTN\_APX" Funktion geprüft. Sie entspringt einem Pointer Toolkit, den ich vor einiger Zeit schon schrieb, und sie gibt eine 1 zurück, wenn ein Job mit dem gegebenen "approximativem" Namen als Button aufgesetzt ist.

Wenn alle meine Programme als Buttons erschienen sind, wird Xchange gestartet, und, nachdem einige Augenblicke gewartet wurde bis es auch richtig da ist, wird der oben erwähnte Hotkey ausgeführt, damit Xchange das kleine TSL Programm lädt und ausführt. Danach wird dieser Hotkey auf F2 gesetzt und eingespeist (damit in Quill das Hilfsfenster verschwindet) und dann wird Xchange, über das QPAC2 "Button\_Sleep" Programm, hinter einem Button schlafen gelegt.

Nun werden zwei Hotkeys aufgestellt, einer ('B') um das Hauptbasic (Job 0) aufzurufen, und der andere ('b') um einen SBasic Tochterjob zu starten. Bis jetzt läuft ja noch alles in Job 0. In Zeile 1380 wird in den Systemvariablen herumgePOKEt, damit die Geschwindigkeit, mit welcher die Tastatur anspringt, etwas erhöht wird, und auch die Wiederholungsrate - damit geht es viel schneller, sich innerhalb einer Linie mit den Cursortasten zu bewegen.

Durch die letzten Zeilen werden einige Fensterwerte über einen "WSET\_DEF" Befehl eingege-

ben, auch wird der Umriß ('outline') von Fenster#0 des Job 0 gesetzt. Dieser Job 0 wird dann als Button schlafen gelegt (Zeile 1440) und ruft einen SBasic Tochterjob auf, der dann mein Haupteingabefenster wird. Dies geschieht um zu verhindern, daß eine gerade geschriebene Extension die ich lade, Job 0 crashen läßt. So crasht halt nur der Sbasic Tochterjob, wenn ich eine Extension lade und ausprobiere... Dieser SBasic Job lädt sich noch selbst ein kleines Programm ein, das aber nur die Fenster auf eine mir angenehme Größe setzt. So endet also mein Bootvorgang, er hinterlässt eine Spalte Buttons und ein großes Basicfenster.

Das Boot-Programm enthält auch einige wenige Prozeduren die noch nicht erwähnt wurden, vor allem die Prozeduren "verif" (Zeilen 2140 bis 2410) und "make\_file" (Zeilen 1080 bis 1960). Die "make\_file" Prozedur erstelle die lange "boot\_rext" Datei die ich oben erwähnte:

Sie liest die Extensions-Dateien in den Speicher, eine hinter der anderen, und speichert dann das Ganze mit einem SBYTES in einer einzigen Datei ab. Die "verif" Prozedur überprüft, ob die Dateilänge von den einzelnen Dateien auch noch richtig ist. Wenn nicht, wird der Sündenbock genannt, und auch die korrekte Länge der Gesamtdatei wird angezeigt, damit ich das einfach in der "tot\_len" Variablen ändern kann.

Die "time" Prozedur (Zeilen 2790 bis 2840) erstellt eine Zeichenkette die das Datum im Format "TTMM.JJJJ" (also Tag Monat Jahr, z.B. 01.03.1999) enthält, und einen Hotkey der "Paris, le " und dieses Datum eintippt. Sie benutzt die "make\_date\$" und "make\_all\_months\$" Funktionen. Es handelt sich da um zwei meiner Standardfunktionen, die Zeitdaten behandeln. Sie mögen etwas umständlich erscheinen. Das Ziel ist, eine Zeichenkette im "TTMM.JJJJ" Format zu bekommen - die Schwierigkeit

liegt in dem "MM" Stück davon. Sie wissen ja, daß die QL DATE\$ Funktion den Monat als eine Abkürzung mit drei Buchstaben (Jan, Feb usw.) produziert. Wie wird das in eine Zahl (01, 02, 03 usw...) umgesetzt? Das hört sich einfach an: Es genügt eine Monatszeichenkette mit den drei ersten Buchstaben aller Monate zu machen:

"JanFebMärAprMaiJunJulAug  
SepOktNovDez" und über die INSTR Funktion herauszufinden, wo die Monatsabkürzung in dieser Zeichenkette liegt. Mit der Formel  $res=(res+2)/3$  (wobei res der von INSTR zurückgegebene Wert ist) bekommt man dann die Zahl des Monats. Genau dies ist auch was die "make\_date\$" Funktion macht. Eigentlich ganz einfach, wäre da nicht ein Problem: Wenn die Sprache des Benutzers nicht Deutsch ist, ist die obige Monatszeichenkette falsch: "Feb" auf Deutsch wird zu "Fäv" auf Französisch, und wenn man in der obigen Monatszeichenkette nach "Fäv" suchen würde,

## Wir suchen freie Programmierer zur Erstellung von kundenspezifischen Applikationen unter SMSQ für eine Betriebs-Datenerfassung

Es handelt sich im Speziellen um serielle Treiber-Anbindungen für Industriekomponenten (nicht auf Assembler-Ebene sondern unter BASIC). Interessenten melden sich bitte unter nebenstehender Adresse oder per Email an [vogel-isernhagen@t-online.de](mailto:vogel-isernhagen@t-online.de)

**VOGEL**

**ANTRIEBS- UND  
STEUERUNGSTECHNIK**

- Planung und Projektierung von Maschinensteuerungen
- Programmierung von Steuerungssystemen (SPS, etc.)
- Anwendungsspezifische Antriebslösungen
- Inbetriebnahme, Service, Instandhaltung



**CONTROL TECHNIQUES** Servicestützpunkt

Joachim Vogel • Hannoversche Str. 22 • 30916 Isernhagen  
Tel. 05 11 / 6 10 57 - 11 • Fax 05 11 / 6 10 57 - 23

würde INSTR das natürlich nicht finden, und 0 zurückgeben! Also muß der Rechner zuerst einmal die Monatszeichenkette selbst erstellen - das ist dann ja in der Sprache die der Benutzer eingestellt hat. Dies geschieht mittels der "make\_all\_months\$" Funktion, welche der DATA\$ Funktion in einer Schleife simulierte Parameter für alle Monate des Jahres gibt, und dann die Monatsabkürzungen daraus herausfiltert...

So, das ist das Ende meines Boot-Programms. Es enthielt auch noch einen Abschnitt um eine Sicherheitskopie von einigen Dateien zu machen. Ich muß Datensicherheit ziemlich ernst nehmen, da ich Buchhaltungs- und andere Daten auf dem Rechner halte. Heute aber sind meine Hauptrechner PCs (unter SMSQ/E mit QXL oder QPCI). Meine QL "Festplatte" ist 70 MB lang. Um davon eine Kopie zu machen, kopiere ich einfach die "QXL.WIN" Datei auf eine ZIP-Kassette. Dafür brauche ich ungefähr 2 Minuten, und am Ende habe ich alle meine Daten kopiert!

Danach brauche ich nur den umgekehrten Vorgang auf den anderen Rechnern vorzunehmen, um die Daten auf sovielen Rechnern wie möglich zu verteilen, damit jeder Datenverlust ausgeschlossen ist. Ich gebe sogar die ZIP-Kassette ab und zu an einen Freund zur Aufbewahrung (er hat keine QL oder ähnliches, kann damit also nicht anfangen) nur um dem Fall vorzubeugen, daß mein Haus und Büro zugleich abbrennen sollten, mit allen Rechnern darin. Ich sagte ja, ich habe einen Sicherheitsfimmel!



## Der QemuLator in der Praxis

*Gerhard Plavec*

Eines meiner letzten Projekte bestand darin, jene - für mich sehr wertvollen - Daten, die ich in Jahrzehnten mit Hilfe meines Lieblings QL-Genealogie-Programms von Chris Boutal gesammelt habe, in eine Form zu bringen, daß sie von Jedermann auf einem jeden Computer problemlos nachzuschlagen sind.

War ich vor zehn Jahren noch der einzige in meiner Verwandtschaft, der mit einem Computer ausgestattet war, so verfügen heute die meisten Familien über einen PC. Nun könnte man mich fragen, warum denn niemand außer mir zu einem QL gegriffen hat, mir vorwerfen, ich hätte den QL nicht ausreichend beworben und angepriesen.

Allerdings sind die Anforderungen, die die Leute heutzutage einem Computer stellen, weit von dem entfernt, was der QL in Stande ist zu leisten. Der QL ist sicherlich ein sehr gutes Gerät, wenn man an der Hardware selbst herum basteln will, er ist sogar ausgezeichnet dafür geeignet, Programme jeder Art zu erstellen, Problemlösungen selbst auszutüfteln - ebenso wenn es darum geht, von Computern wirklich etwas zu verstehen - und natürlich Programme absturzsicher laufen zu lassen.

Was heute jedoch gefordert wird, ist ein Gerät, mit dem vor allem Ahnungslose leicht zu recht kommen, mit möglichst vielen bunten Farben und musik-ähnlicher Geräuschwiedergabe ausgestattet, für das nicht das kleinste Programm zu schreiben ist, weil alles schon fix und fertig vorhanden, dazu noch leicht und möglichst umsonst zu bekommen ist. Und dann sind da noch diese unzähligen schön-schauerhaften Spiele...

Weiters wurden auch meine Verwandten mit den Problemen

der Arbeitssuche, Wiedereinstieg nach Kindererziehung und Arbeitslosigkeit konfrontiert, wo überall stets Computererfahrung verlangt wird - wobei unter Computererfahrung eben nicht QL-Praxis, sondern der Umgang mit MicroSofts Zeug gefragt ist - sogar von Unternehmen, die mit Macs ausgestattet sind!

Nun habe ich mir überlegt, in welcher Form ich meine Daten aufbereiten sollte. Ein bestimmtes Genealogie-Programm stand mir nicht zur Verfügung bzw. die angebotene Freeware entsprach nicht meinen Erwartungen (einmal davon abgesehen, daß so ein Programm kaum Dauerhaftigkeit erweist und keineswegs gesichert ist, daß es weiterentwickelt wird). Nach meiner Rechnung sollte es ja bereits heute MicroSoft und Intel schon lange nicht mehr geben ;-)

Am günstigsten schien es mir demnach, meine Daten ins HTML-Format zu übertragen (das Internet dürfte doch eher Bestand haben) und mit Links zu den verschiedenen Ahnen bzw. Nachkommen auszustatten. Eigentlich ist das HTML-Format wie geschaffen für Genealogie...

Nachteil: Features wie den "gesamten Stammbaum zeichnen" oder "wer ist mit wem wie verwandt" u.ä.m. fallen da natürlich unter dem Tisch, aber eines Tages kann man das ja vielleicht mit einem kleinen Java- oder Perl-Programm nachholen.

Der Vorteil des HTML-Formates ist wiederum, daß man sehr leicht jede Menge Bilder, Töne und Videos nebst Scanns von Originaldokumenten hinzufügen kann. In meinem Umwandlungsprogramm habe ich gleich Platz für vier Porträt-Bilder vorgesehen - sozusagen für die vier Jahreszeiten des Lebens eines jeden Menschens.

Das Programm werde ich hier weder weiter beschreiben noch abdrucken, da es doch recht umfangreich und nicht für jedermann brauchbar ist. Sollte

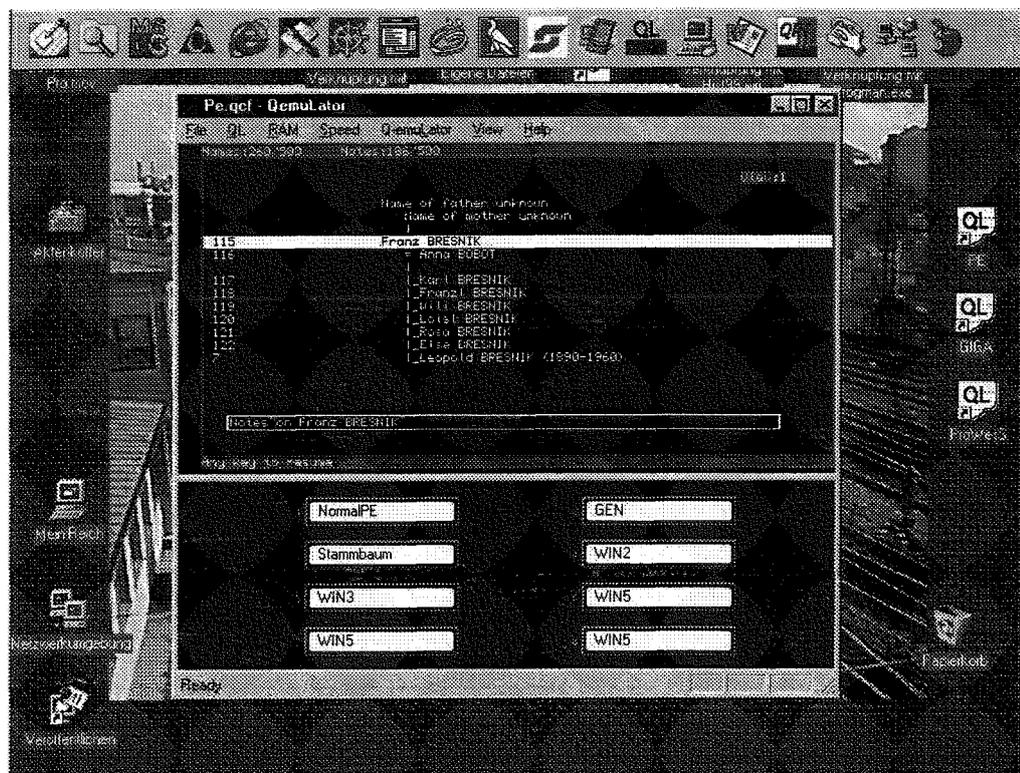
ser) herzustellen.

Sowohl die Lösung mit vernetzten Rechnern - geschweige denn jene langwierige, mit "mir" als hauptberuflichen Diskjockey - als auch das permanente Herunter- und Hinauffahren abwechselnd von QPC und Windows, wollte mir nicht so recht schmecken. Somit kam eigentlich nur noch QPC2 von Marcel Kilgus und der QemuLator von Daniele Terdina in Frage. Beide Emulatoren laufen in Multitasking unter Windows. Da mir jedoch QPC2

name\_Geburtsjahr besteht. Nur vor einer allzu liberalen Verwendung von QL-Subdirectories möchte ich hier warnen; zumindest sollten sie besser aus der Windows-Ebene erzeugt werden. Zusätzlich erhielt vor Kurzem der Emulator eine wichtige Verbesserung, die es ermöglicht, auf QXLWIN-Files direkt - wenn auch nur lesend - zuzugreifen. Somit stellte sich heraus, daß sich dieser Emulator - der bei mir sowohl mit QDOS als auch Minerva fehlerfrei läuft - hervorragend für mein Unterfangen eignete.

Die etwas ältere Mac-Version des QemuLators hätte sich vermutlich genauso bewährt, allerdings fehlt ihr die Eigenschaft auf QXLWIN-Files zuzugreifen zu können, welche der Autor angibt, auch in Zukunft nicht einbauen zu wollen. Na ja, hoffentlich ist da noch nicht das letzte Wort gesprochen.

Sehr gut gefällt mir die übersichtliche Zuweisung der einzelnen Laufwerke. Ich könnte mir



jedoch Interesse daran bestehen, könnte ich es im Internet oder für eine QL-Today-Diskette zur Verfügung stellen. Grundsätzlich wäre jeder QL geeignet gewesen, die Daten umzuwandeln, da ich jedoch die HTML-Sprache noch nicht so recht beherrsche und um eine Art WYSIWYG (was-Du-siehst-ist-was-Du-kriegst) zu ermöglichen, war es mir wichtig, ein möglichst großes Näheverhältnis zwischen QL und PC (mit HTML-Viewer bzw. Brow-

derzeit lediglich als Demo-Version ohne Schreibzugriff auf Festplatte und Diskette vorliegt, hielt sich die Qual der Wahl schließlich doch eher in Grenzen.

Der QemuLator bindet bereits von Haus aus, das Windows-Filesystem voll ins QL-System ein und erlaubt somit unter Win95/98 lange fantasievolle Filenamen - z.B. habe ich für jeden meiner Verwandten ein HTML-File erzeugt, dessen Namen aus Vorname\_Familien-

allerdings vorstellen, daß eine Angabe der Windows-Pfade (vielleicht noch zwischen den Zeilen) doch sehr hilfreich sein könnte (derzeit muß man sich - etwas umständlich - das QCF-File ansehen).

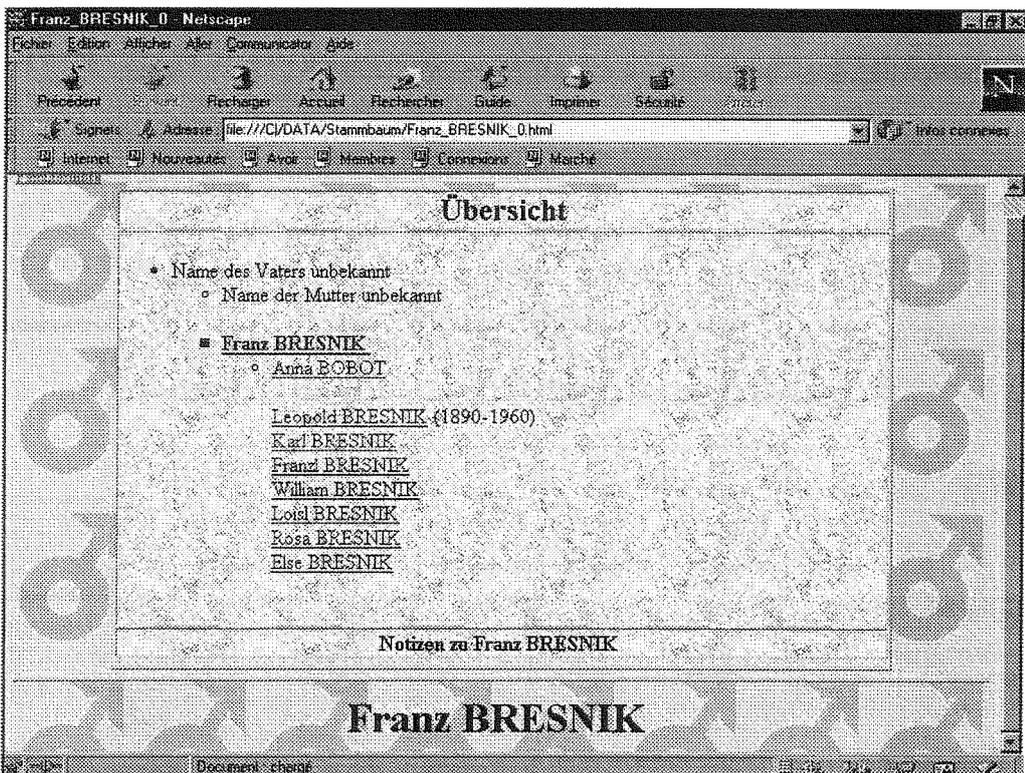
QPCx erlauben es nun ja auch, die Belegung der "WINx\_" aus der QL-Ebene sowohl festzulegen (z.B. WIN\_DRIVE 5,"D:\QPC\WIN5\QXL.WIN") als auch abzufragen (z.B. FOR n=1 TO 8 :PRINT WIN\_DRIVE\$(n)) - für all jene QPC-Besitzer wohl

auch eine Überraschung, die die Handbücher schon lange nicht mehr lesen ;-) Wünschenswert wäre nun sowohl bei QemuLator als auch QPC2 eine Möglichkeit vorzufinden, den Windows Zwischenspeicher mitzubeneützen. Zumindest in der Richtung Windows -> QL sollte es nicht allzu schwierig sein, sofern es sich um Text handelt. Weiters wären natürlich PIPEs zwischen QL- und Windows-Programmen bzw. zwischen

der diversen Editoren fragen, ob sie sich nicht überlegen könnten, ihr Werk mit einer HTML-Ausgabe zu versehen - wäre eine echte "über-drüber" Bereicherung! Als weitere QL-Emulatoren, die unter fremden Systemen mehr oder weniger in Multitasking laufen, gibt es da noch die QLayer-Trilogie (unter DOS, Windows und Linux) sowie SPEM (unter Atari-Systeme). Abgesehen davon, daß ich in früheren Versuchen erhebliche

sogar kategorisch meine Existenz! Wie es wohl dazu kam? Für einen aufrechten QL-User kaum vorstellbar: Ich muß fairerweise zugeben, daß ich sehr wahrscheinlich selbst an dieser Misere Schuld tragen dürfte. Möglicherweise hätte ich mich als blutiger Anfänger und Laie nicht als Sysop anmelden dürfen - aber so ein hartgesotterer QL-User, wie ich einer zu sein glaube, verträgt es halt kaum, nicht als "Herr" über seinen Computer zu herrschen. Und dann hätte ich vielleicht auch nicht einfach und unbekümmert in der Datei "/etc/passwd", die die Anmeldeberechtigungen enthält, wild umhereditieren dürfen - dabei ging ich doch eh ganz behutsam vor. Die Linux-Editoren sind jedoch alles andere als das, was man als verwöhnter QL-User gewöhnt ist. Nun bootet Linux zwar brav und kommt auch anstandslos bis zur Anmeldeaufforderung. Dann allerdings gibt's Saueres: Jeder Anmeldeversuch wird mit der Meldung "you don't exist, go away" quittiert und dann erkennt Linux nicht einmal die Tastenkombination CTRL-ALT-DEL bzw. STRG-ALT-ENTF mehr, so daß ich den Computer nur noch abschalten kann.

Na ja, ist wohl wieder ein Tag für's Hinbiegen fällig - man hat ja sonst nichts zu tun. ■



QL-Programmen, die am QemuLator einerseits und QPC2 andererseits laufen (schließlich wäre das eine Möglichkeit, die Vorteile von QDOS- bzw. Minerva- einerseits mit jenen von SMSQ/E-Programmen andererseits zu verbinden. QDOS bzw. Minerva erlauben es zum Beispiel ältere Programme, die sich mit dem Pointer-Environment nicht vertragen, ohne Änderungen laufen zu lassen. Wenn ich schon beim Wünschen bin, möchte ich bei der Gelegenheit auch alle Autoren

Mängel feststellen mußte, wollte ich keine weitere Zeit dafür aufwenden, die Grenzen dieser Emulatoren aufzuspüren. Grundsätzlich sollte selbstverständlich alles, was ich über den QemuLator geschrieben habe, auch für uQLx - den QL-Emulator unter Linux - gelten, da dieser vom QemuLator abgeleitet wurde. Allerdings hatte ich keine Chance es auszuprobieren, da es in letzter Zeit Linux ablehnt, sich mit mir auch nur zu unterhalten - schlimmer noch, es negiert

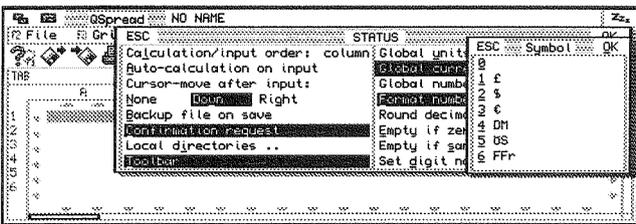
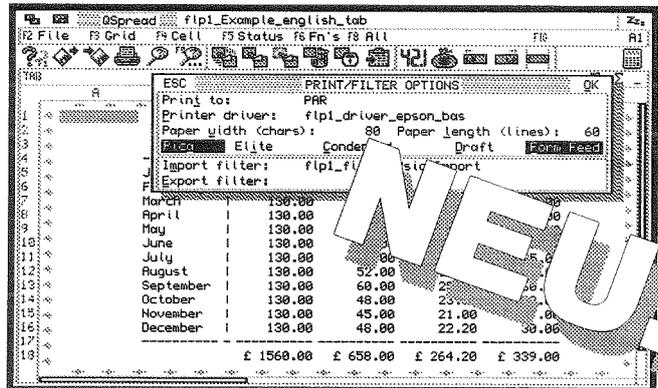
Im stillen Winkel 12 • 47169 Duisburg • Germany  
 ☎ 0203-502011 (Fax 0203-502012 Mailbox 0203-502013 & 502014)  
<http://www.j-m-s.com/smsq/> email: smsq@j-m-s.com

# QSpread99

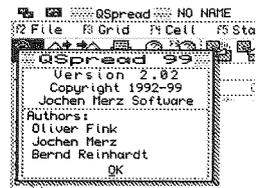
Endlich ist es vollbracht: Eine ganz neue Version von QSpread existiert! Wir haben viel Zeit investiert und QSpread stark aktualisiert und zudem noch eine Menge Bugs herauskorrigiert.

### Neue Fähigkeiten:

- QD-ähnliche Werkzeugleiste mit vielen neuen Funktionen.
- QD-ähnliche Menüs, Abkürzungen und Kurzwahltafeln.
- SHIFT TAB für sehr lange Formelzeilen-Eingabe.
- ALT TAB bringt den Mauszeiger außerhalb des Gitters - wenn Sie keine Maus haben.
- ESC legt QSpread schlafen.
- QDs Druckertreiber können nun auch für QSpread genutzt werden - das gesamte Drucken wurde neu geschrieben und stark vereinfacht/verbessert.
- Deutlich bessere Sicherheits-Abfragen.
- Alle Währungs-Symbole sind nun konfigurierbar, so auch der EURO.
- CNTNUM() berechnet nur die numerischen Felder innerhalb eines Bereichs.
- Mehrfache "-" und "====" werden nun automatisch als Text interpretiert.



Manche Sonderzeichen wie "|" werden auch als Text interpretiert.  
 Automatische Cursor-Bewegung ist nun konfigurierbar, kann auch ausgeschaltet werden.  
 Neue DATUM-Makro-Funktion, die das aktuelle Datum ohne Tipperei einfügt.  
 Das Datums-Format ist konfigurierbar, kann auch im US-Format und mit 2/4-stelliger Jahreszahl eingefügt werden. Auch das Datums-Trennzeichen ist einstellbar.



Außerdem wurde in QSpread einige länger bekannte Bugs beseitigt, so z.B.:  
 Fenstergröße kann verändert werden nachdem QSpread schlafen gelegt wurde.  
 DATE\$ zeigt auch Daten nach dem Jahr-2000 korrekt an.  
 Zwei "." in Zahlenwerten, Konvertierungsprobleme von kleinen Zahlenwerten und andere Probleme sind beseitigt. Viele Fehler im Formelparser wurden beseitigt..

Dies sind die größeren Änderungen - es gibt zudem noch eine Menge kleinerer Verbesserungen. Zudem gibt es eine neue Anleitung, nicht nur Update-Seiten.

**Der Upgrade-Preis beträgt DM 39,90. Bitte senden Sie die Original-Diskette für ein Upgrade zurück.  
 Ein neues QSpread99 kostet nur noch DM 149,- PREISREDUKTION!**

### LIEFER- und ZAHLUNGSBEDINGUNGEN

Versandkosten [Deutschland] DM 8,99 (wenn Rechnungsbetrag unter DM 50,- dann nur DM 5,99). Bei Rechnungsbeträgen über DM 500,- kostet es

DM 18,99 [Europa] DM 14,50 (wenn Rechnungsbetrag unter DM 50,- dann nur DM 9,50). Alle Preise inkl. 15% MwSt. Irrtum und Preisänderung vorbehalten. Verrechnungs-, Eurochecks und Kreditkarten werden akzeptiert. Bankeinzug möglich.



# Wieder einmal BASIC

Helmut Boll

Start einer Reihe kleiner BASIC-Programme, angefangen beim BOOT. Ja, das zweite BOOT in dieser Ausgabe, doch handelt es sich gerade beim BOOT um ein sehr interessantes Thema - jedes BOOT ist anders und aus jedem kann man für sein eigenes BOOT etwas lernen.

Wieder einmal Basic!? "Warum denn nicht?" werden die Einen sagen, wenn's was interessantes verspricht. "So was Ödes" werden die Anderen sich äußern, "das habe ich mir schon längst in den Kinderschuhen abgelaufen." Ich selbst zähle mich zu der ersten Gruppe. Nicht nur weil BASIC mir gefällt bzw. Spaß macht sondern vor allem, weil man sich Programme, die man benötigt oder zumindest einmal zur Verfügung haben möchte, selbst erarbeiten kann ohne gleich ein gewiefter Programmierer zu sein. Der besondere Reiz liegt für mich jetzt darin, 'QPC' und 'SMSQ/E' im PC nutzen zu können, im Bildschirmformat 640 x 480 Pixel. Leider ist die Programmvielfalt meines Wissens gleich Null. **[Also, ganz so schlimm ist es ja nun doch nicht, oder? - Editor]** Also bietet sich doch die Selbsthilfe förmlich an. Hier möchte ich jetzt mit einer kleinen Serie von kleinen Basicprogrammen zum selbsttippen Anregungen zur Abhilfe geben. Der größte Aufwand sind lediglich die ersten beiden (Boot und Basis) Programme. Bei allen weiteren, folgenden Programmen sind dann nur noch ein Paar Titelzeilen, Fensterabmessungen sowie die Ein- und Ausgabe-prozeduren zu ergänzen bzw. hinzuzufügen. Die Programme sollen durch die Einfachheit nicht zur Last werden und nur etwas Freude erwecken. Um nun aber auch nicht lästig zu werden, werde ich die Folge spätestens nach dem dritten

Programm abbrechen, falls nicht der Wunsch nach 'mehr' laut wird. Jochen Merz wird das Echo gerne registrieren. Nun zum ersten Teil - das Bootprogramm:

- Es wird zweckmäßig über FLP1\_bereitgestellt.
- Es stellt nach dem Start den Bildschirm auf 640 x 480 Pixel um und definiert in der Prozedur 'Stamm' die bekannten Fenster #0, #1, und #2. Nebenbei werden die Fenster #13 und #14 zur ständigen Zeitanzeige geöffnet.
- Hier im Bootprogramm ist es nun natürlich möglich, die unterschiedlichsten Hilfs-

programme oder Prozeduren unter zu bringen. Ich habe z.B. eine Prozedur zum Aufruf von QPAC2 und eine zum Aufruf der QL-Farben untergebracht. Siehe Listing 1 "BOOT".

Die Prozedur 'Lade' wird später durch ein neues Bootprogramm mit 'AUTODIR' überflüssig, welches dem Auge auch etwas mehr als das einfache DIR bietet.

Das Basisprogramm: Bei allen Programmen auf dieser Basis habe ich Wert auch auf eine entsprechende Ansicht gelegt. D.h.: Der Bildschirm wird in einzelne Fenster aufgeteilt, die vorwiegend mittig angeordnet werden. Einzugeben sind nur die Breiten br6, br8 und br10, und die Höhen hö6, hö8 und hö10 der Fenster (Zeilen 290 bis 294). Die Schattenbreite kann nach eigenem Ermessen durch die Variable 'st' in Zeile 288 geändert werden.

## Listing 1 - BOOT

```
20 REMark Boot zum Basisprogramm passend
22 DISP_SIZE 640,480
24 MODE 8
26 Stamm :DIR: Lade
28 :
30 DEFine PROCedure Stamm
32 WINDOW #1,640,438,0,0:PAPER 220:CLS
34 BORDER #1,2,4:INK 0:CSIZE #1,2,1
36 WINDOW #2,636,434,2,2:INK#2,0:CSIZE#2,0,0
38 WINDOW #0,528,42,0,438
40 BORDER #0,1,4
42 INK #0,7:CSIZE #0,2,0:CLS #0
44 OPEN #13,scr_112x42a528x438
46 BORDER #13,3,4
48 OPEN #14,scr_95x20a540x450
50 CSIZE #14,2,1
52 CLOCK #14,'%h : %m'
54 HOT_GO
56 END DEFine
58 :
60 DEFine PROCedure SA
62 SAVE_O flp1_boot
64 END DEFine SA
66 :
68 DEFine PROCedure Lade
70 PAPER #0,2:CLS #0
72 INPUT #0;"Welches programm starten ? ";A$
74 LRUN 'FLP1_&A$
76 END DEFine
```

Programmgrundsätze:

- Die Prozedur 'Stamm' bleibt im Grunde wie im BOOT erhalten und wird nur zum Erneuern des CLOCK-Befehls mit aufgerufen.
- Ferner hat jeder die Möglichkeit die Grundfarbe (#1) dem eigenen Geschmack und Empfinden anzupassen. Ein entsprechendes Farbprogramm steht in dieser Reihe etwas später zur Verfügung.
- Das Basisprogramm sollte nicht umnummeriert werden,

sodaß die Verständigung zu jeder Zeit möglich ist. Also Start bei Zeile 200, Ende unbedingt unter 500, damit das eigentliche Programm immer in Zeile 500 beginnen kann.

- Es stehen drei Titelzeilen zur Verfügung von denen bei Nichtbenutzung der dritten die zu c\$ gehörenden Zeilen natürlich auch gelöscht werden dürfen. In diesem Fall kann die Variable y6 in Zeile 288 auf 50 verringert werden. Der frei

werdende Platz steht damit für die anderen Fenster zur Verfügung.

- Die Prozeduren 'CONT', 'ENDE' und 'SA' bleiben bei jedem neuen Programm in der vorliegenden Form erhalten.

Siehe Listing 2 - "Basis"

Nun wünsche ich viel Erfolg beim Eintippen. Im nächsten Heft wird dann das Programm 'Wochentag' durch das Ergänzen mit der Prozedur 'Proz' entstehen.

## Listing 2 - BASIS

```
200 REMark Basisprogramm
202 Stamm:wind:Titel
204 :
206 DEFine PROCEDURE Stamm
208     WINDOW #1,640,438,0,0:PAPER 220:CLS
210     BORDER #1,2,4:INK 0:CSIZE 2,1
212     WINDOW #2,636,434,2,2:INK#2,0:CSIZE#2,0,0
214     WINDOW #0,528,42,0,438
216     BORDER #0,1,4
218     INK #0,7:CSIZE #0,2,0:CLS #0
220     OPEN #13,scr_112x42a528x438
222     BORDER #13,3,4
224     OPEN #14,scr_95x20a540x450
226     CSIZE #14,2,1
228     CLOCK #14,'%h : %m'
230     HOT_GO
232 END DEFine
234 :
236 DEFine PROCEDURE Titel
238     a$="Dieses Programm berechnet .... u.s.w"
240     b$="Dies sind die Titelzeilen!"
242 REMark c$="Dies ebenfalls noch"
244     CURSOR #1,(636-LEN(a$)*12)/2,2:PRINT;a$
246     CURSOR #1,(636-LEN(b$)*12)/2,23:PRINT;b$
248 REMark CURSOR #1,(636-LEN(c$)*12)/2,44:PRINT;c$
250 END DEFine
252 :
254 DEFine PROCEDURE CONT
256     LOCAL au
258     CSIZE #12,3,1: CURSOR #12,8,2
260     PRINT #12;'Programmende ';FILL$(CHR$(189),3);' ESC '
262     CSIZE #12,2,0: CURSOR #12,8,28
264     PRINT #12;'Wiederholung ';FILL$(CHR$(189),5);' Space '
266     REPEAT loop
268         au=CODE(INKEY$(-1))
270         SELECT ON au
272             =27:ENDE: STOP
274             =32:PROZ:CONT
276         END SELECT
278     END REPEAT loop
280 END DEFine
282 :
284 DEFine PROCEDURE wind
286     LOCAL i
288     scx=640 :      scy=438      :y6=72      :st=6
```

```

290 br6=480 : h6=95
292 br8=580 : h8=65
294 br10=620 : h10=9
296 br12=340 : h12=50
298 abst=(scx-y6-h6-h8-h10-h12)/4
300 y8=y6 +h6 +abst
302 y10=y8 +h8 +abst
304 y12=y10+h10+abst
306 FOR i=5 TO 13:OPEN #i,'scr_':OPEN #6,'con_'
308 REMark Definition der Schatten
310 WINDOW #5,br6,h6,(scx-br6)/2+st,y6+st
312 WINDOW #7,br8,h8,(scx-br8)/2+st,y8+st
314 WINDOW #9,br10,h10,(scx-br10)/2+st,y10+st
316 WINDOW #11,br12,h12,(scx-br12)/2+st,y12+st
318 FOR i=5,7,9,11: PAPER #i,0:CLS #i
320 REMark Definition des Inputwindow
322 WINDOW #6,br6,h6,(scx-br6)/2,y6
324 PAPER #6,219:CLS #6:INK #6,7
326 BORDER #6,2,3: CSIZE #6,2,1
328 REMark Definition des Outputwindow
330 WINDOW #8,br8,h8,(scx-br8)/2,y8
332 PAPER #8,5: CLS #8: INK #8,0
334 BORDER #8,2,1: CSIZE #8,3,1
336 REMark Definition des Kommentarwindow
338 WINDOW #10,br10,h10,(scx-br10)/2,y10
340 PAPER #10,200:CLS #10: INK #10,206
342 BORDER #10,3,202
344 REMark Definition des Auswahlwindow
346 WINDOW #12,br12,h12,(scx-br12)/2,y12
348 PAPER #12,227:CLS #12: INK #12,0
350 BORDER #12,3,2
352 END DEFine
354 :
356 DEFine PROCedure ENDE
358 CLS
360 AT 20,14:PRINT;"A d i e u !"
362 FOR n=5 TO 65 STEP 5:SCROLL -n
364 END DEFine
366 :
368 DEFine PROCedure SA
370 SAVE_0 flp1_Basis
372 END DEFine

```

## Darf ich mal kurz unterbrechen...?

Jochen Merz

Dieser Text richtet sich in Hauptsache an alle Programmierer, sei es für Anwendungen die für sich selbst gedacht sind oder auch für Anwendungen, die für Dritte geschrieben wurden.

SMSQ/E hat ja bislang keine "normale" Möglichkeit, ein laufendes Programm "schön" "von außen" zu beenden, ebenso können zeitintensive Aktionen, nachdem sie gestartet wurden, kaum mehr abgebrochen werden.

Zu den vielen Anführungszeichen: "normal" soll heißen, ohne herumzuPOKEn oder sonstwie unsaubere Wege zu nutzen.

"schön" beenden heißt, das Programm soll von sich aus vernünftig herunterfahren, wenn es dazu in der Lage ist. Ein RJOB oder LJOB entfernt das Programm ja unter allen Umständen. Dabei ist ganz egal, ob der bearbeitete Inhalt des Programms gespeichert ist, ob es sich mitten in einer Schreib-Operation befindet oder wie auch immer.

"von außen" heißt, wir wollen in ein Menü klicken und dadurch ein anderes Programm zum Beenden veranlassen oder eine langwierige Aktion abbrechen; dies soll natürlich auch durch Befehlseingabe gehen.

Konkrete Beispiele: Wenn man viele Editoren, Tabellenkalkulationen und dergleichen gleichzeitig laufen läßt und den Rechner ausschalten will, dann sollte man allen laufenden Programmen mitteilen können: "beende dich, doch wenn du noch was zu erledigen hast (z.B. geänderten Text

speichern), dann erledige das mal zuerst". Oder das "unterbrechen": Wer hat nicht schon einen gesamten Baum der Festplatte in QPAC2 anlegen lassen, obschon man es gar nicht wollte, mit Sortierung dauert das unter Umständen mehrere Minuten. Wäre hier ein "Abbrechen" nicht schön? Oder eine fälschlicherweise Zwangs-Assemblierung eines gigantischen Projekts abbrechen können wäre ja auch nicht schlecht.

Wir haben hier eine Idee, die wir gerne mit allen Programmierern diskutieren möchten und erwarten hier möglichst bald Reaktionen. Bitte nehmt euch die paar Minuten und antwortet, ja?

Es ist klar, daß die Programme auf die beiden oben genannten Ereignisse nur dann reagieren können, wenn es ihnen paßt. Die Art der Signalisierung ist daher schnell gewählt: Job-Events! SMSQ/E bietet ja nun schon seit geraumer Zeit Job-Events, die von jedem Programm genutzt werden können, so auch von BASIC. Wir haben acht Bits zur Verfügung, und die sollten gut

durchdacht genutzt werden - vielleicht will der eine oder andere Programmierer sie ja auch für eigene Zwecke nutzen.

Daher unser Vorschlag: wir einigen uns auf zwei Bits (z.B. 6 und 7). Bleibt zu hoffen, daß sie noch nicht benutzt sind. Wer sie nutzt ist hiermit gefragt, ob es möglich ist, sie freizubekommen und andere Bits zu nutzen bzw. Vorschläge für andere Bits zu bringen - ich koordiniere dies dann gerne.

Die Jobs sollten dann bei langwierigen Aktionen in regelmäßigen Abständen die Job-Events abfragen und dementsprechend reagieren. Wie das geht - in Maschinencode und in BASIC - erkläre ich gerne, würde aber erst mal Reaktionen zu den Vorschlägen erhalten. Also, überdenkt's mal und schreibt - am einfachsten per Email an [SMSQ-Events@j-m-s.com](mailto:SMSQ-Events@j-m-s.com)



---

## Die Cover-Disk

Einige der Dateien auf der Diskette sind mit einem Programm namens ZIP komprimiert damit wir mehr auf die Disk packen konnten.

### WICHTIGER HINWEIS

Die Diskette enthält Dateien die mit Info-ZIP komprimiert wurden. Diese Dateien müssen mit dem auf der Disk befindlichen UNZIP-Programm entpackt werden. Info-ZIP's Software (Zip, Unzip und andere Utilities) ist kostenlos und kann auch als Source-Code oder fertige Datei aus dem Internet/WWW bezogen werden: <http://www.cdrom.com/pub/infozip/>

Sie können auch Versionen für QDOS/SMS auf Jonathan Hudson's Dead Letter Drop Webseite: <http://www.jrhudson.demon.co.uk>

### UNZIP VERWENDEN

Sie können die vollständige Anleitung auf den oben genannten Webseiten erhalten. Nur kurz, UNZIP entpackt Dateien auf das mit DATA\_USE eingestellte Gerät. Um beispielsweise Paragraph, das sich auch auf der Disk befindet, zu entpacken, schieben Sie die QL Today Cover Disk in FLP1\_ und eine frisch formatierte Disk in FLP2\_ (Sie können auch RAM1\_ an Stelle von FLP2\_ nehmen wenn Sie nur ein Laufwerk besitzen), dann tippen Sie in SuperBASIC oder SBASIC:

```
DATA_USE FLP2_
```

```
EX FLP1_UNZIP; 'FLP1_PARA104_ZIP'
```

Auf dem Bildschirm erscheint nun eine Liste der Dateien die gerade entpackt werden. Warten Sie bis UNZIP fertig ist, es kann ein paar Minuten dauern. Danach finden Sie alle entpackten Dateien auf Laufwerk 2.

## Die Dateien

**dbMONITEST\_ZIP** Testbilder für Monitor und Fernseher

**dbERINNERN\_ZIP** Anzeige von Geburtstagen und Veranstaltungen, 03/98

Folgende Dateien gehören zusammen, sind ggf. auch einzeln nutzbar:

**dbTRATAB\_ZIP** TRA-Tabellen für Bastel- und Download-Euro, 01/99

**dbQLFNT\_ZIP** QL-Fonts mit Euro und vielen EPSON-Zeichen, 01/99

**dbBOOT\_ZIP** BOOT-Ergänzungen für TRA-Tabellen und QL-Font

**dbEURO\_ZIP** Drucker-Befehle für Download-Euro, 01/99

**dbNAMEN\_ZIP** Datenbank speziell für Namen mit Generierung von Anschriften für Textverarbeitungen, 07/98 und 09/98

**BOOT\_bas** Listing aus dieser Ausgabe.

**BOLL\_zip** Dateien zum Artikel von H. Boll.

**PARA104\_zip** Aktuelle Version von Paragraph

**GV115\_zip** Graphics Viewer V1.15 von Dilwyn Jones (auf mehrfachen Wunsch)

Zum Vergnügen: Alle Dateien, die mit dem Namen **Spiel\_** beginnen, sind Spiele von F. G. Dibowski.

Zu guter Letzt: **JMSList.txt** enthält alle Dateien die es in der JMS-Mailbox gibt - außer den Telefonkosten fallen keine Gebühren an, also einfach mal einwählen, herunterladen was einem gefällt und ausprobieren...

# QL-Treffen 1999

Auch in diesem Jahr stehen natürlich wieder QL-Treffen auf dem Plan:

## **Samstag, 26. Juni - Eindhoven**

Natürlich wieder im St.-Joris College, gleicher Platz wie immer. Beginn um 10 Uhr, Ende gegen 16 Uhr. Wieder eines der "regulären" Treffen in Eindhoven, Jochen Merz wird dieses Mal natürlich auch wieder dort sein!

## **Samstag/Sonntag, 9.-10. Oktober - Heidenreichstein (Österreich)**

Das zweite Treffen in Heidenreichstein in Niederösterreich, nahe der Grenze zu Tschechien, steht im Herbst an. Wieder im Gasthof-Restaurant "Nöbauer" (Schremser Straße 28, A-3860 Heidenreichstein) - wie letztes Jahr. Zimmerreservierungen sind möglich unter Tel.: 0043(0)2862 52237 oder 52746

Geplante (neben)Veranstaltungen:

Besuch der Schmalspur-Expo 1999 in Gmünd.N.Ö., organisiert von der Arbeitsgemeinschaft Schmalspur. Alles zum Thema Schmalspur in Echt sowie im Modell - also besser gleich vormerken...

Anfahrt mit der Bahn: Von Prag oder Wien über Gmünd, dann mit dem Bus

- mit dem Auto:  
siehe Karte:

