

QL Today

DEUTSCH

Jahrgang 1
Ausgabe 5
Jan./Februar
1997

ISSN 1432-5446

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

ERFAHRUNGSBERICHT LDUMP



... UND EINIGES MEHR!

Herausgeber:

Jochen Merz Software
Im stillen Winkel 12
47169 Duisburg
Deutschland

Tel. +49 203 502011
Fax +49 203 502012
Box1 +49 203 502013
Box2 +49 203 502014

QL Today erscheint alle zwei Monate, Erscheinungsdatum der ersten Ausgabe ist der 15. Mai. Das Abo beginnt mit der aktuellen Ausgabe zum Zeitpunkt des Eingangs des Abos. Das Abo kostet wie folgt:

Deutschland	DM 70,-
England	DM 60,- or £25
Rest der Welt	DM 70,- or £30

Bezahlung kann in DM erfolgen, entweder mit Verrechnungsscheck einer in Bank mit Sitz in Deutschland oder Eurocheck. Schecks sollten auf Jochen Merz Software ausgestellt sein. Es besteht auch die bequeme Möglichkeit der Einzugsermächtigung, auch hier nur bei Banken in Deutschland.

Bezahlung per Kreditkarte ist ebenfalls möglich, entweder durch Anruf oder Einsenden des Formulars auf der letzten Seite. Es werden sowohl die Kartenummer als auch das Verfalldatum der Karte benötigt.

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 platziert werden soll, geben Sie es bitte auch an.

Redaktionsschluß für Artikel und Werbung:

Ausgabe 1: 15. April	Ausgabe 2: 15. Juni
Ausgabe 3: 15. August	Ausgabe 4: 15. Oktober
Ausgabe 5: 15. Dezember	Ausgabe 6: 15. Februar

QL Today behält sich vor, eingeschicktes Material nicht zu veröffentlichen. **QL Today** ist unter keinen Umständen für die Richtigkeit der abgedruckten Artikel und Programme 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.

Inhalt

- 3 Editorial
- 4 Kleine Korrektur
- 4 Wofür braucht man das "Pointer Environment" oder "PE für Anfänger" - Ralf Tenbrink
- 7 MAUS an QXL2 - Carlos Brennecke
- 7 BASICV - Teil 2 - H.P. Recktenwald
- 12 LDump für den HP Deskjet 500 - Helmut Friedrich
- 15 Mein Bootfile - Thomas Menschel
- 17 Verschiedenes - Teil 2 - Friedemann Oertel

Kleinanzeigen

Da **QL Today** eine der wichtigsten Quellen für QL-Neuigkeiten werden wird, 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.

Inserenten

Rückseite - Jochen Merz Software

Editorial

Jochen Merz

Liebe QL Today-Leser,
ich hoffe, Ihr seid alle gut ins Neue Jahr gekommen.

Ich nehme dieses Editorial zum Anlaß, einen Rückblick über 1996 zu bringen. Viel ist geschehen, genauer gesagt, wesentlich mehr als in den letzten zwei Jahren, wenn ich mal so nachdenke:

Eigentlich wurde der QL-Szene durch das neue Erscheinen von QL Today im Mai letzten Jahres wieder neues Leben eingehaucht - wir haben sehr, sehr viele positive Zuschriften erhalten. In Deutschland ist der Anklang leider nicht so groß wie erwartet, aber vielleicht liegt dies ja auch daran, daß sich mehrere Leser eine Ausgabe teilen. Daher ein kleiner Appell an die "Mitleser": wenn ihr QL Today gut findet, abonniert es doch bitte selber. Nur so kann die deutsche Zusatz-Ausgabe auf Dauer gerechtfertigt werden. Bislang ist ein Zeitaufwand von über zusätzlichen 30% für rund 1/5 der Leserschaft nicht kostendeckend.

Wir hatten sehr, sehr viele und schöne QL-Treffen in 1996 - ich will sie jetzt nicht alle durchgehen, denn wer dort war weiß wie prima die Treffen ankommen, und wer nicht gekommen ist, sollte es sich für dieses Jahr dick im Kalender markieren.

Miracle hat wieder sowohl SuperGoldCards als auch QXL2 neu aufgelegt, obwohl es im Vorjahr hieß, es würden keine mehr gemacht und neue Produkte kommen. Nun, die neuen Produkte kommen auch - nur braucht Stuart dazu etwas Zeit (die er sich jetzt konkret nimmt) - QBranch hat jetzt den Miracle Kundenservice übernommen und Stuart hat nun Zeit und Konzentration, etwas auf die Beine zu stellen.

Ich selber merke ja auch, wie wenig Zeit für's Programmieren übrig bleibt mit all dem Schreibkram, Telefon, Bestellungen und so weiter und so fort. Wenn man konzentriert versucht zu programmieren und das Telefon unterbricht einen ständig, dann kommt man nicht sehr weit. Daher ist mein Ski-Urlaub gewöhnlich auch recht produktiv: kein Telefon in der Nähe, und lange Abende die man mit dem Laptop nutzen kann.

Was mich dann auch gleich zum Produkt des Jahres bringt: QPC - der Software-QL-Emulator für PC's. Mit ihm ist der Wunsch vieler QL'er wahr geworden: Nun kann man seine QL-Programme auch auf modernen Laptops laufen lassen. Natürlich läuft QPC auch prima auf normalen PCs, aber er ist nun mal DIE Lösung für Laptops.

Als nächstes herausragendes Produkt haben wir

die Aurora noch kurz vor Ablauf des Jahres erhalten. Nun ist damit die gesamte Hardware des QLs durch aktuelle Bauteile ersetzt - was will man mehr? Die Weichen für die Zukunft sind auch gestellt, da die Aurora auch für 32-Bit Bus und andere zukünftige Erweiterungen bereits vorbereitet ist.

Natürlich gab's auch einiges an Software: ProWesS ist aus dem Beta-Stadium in eine vernünftige Version gewachsen. Neue ProWesS Anwendungen gibt es bereits auch schon. Ein neues LineDesign ist in Arbeit. Für viele war ProWesS bislang noch nicht sehr nützlich, aber das lag in Hauptsache an der geringen Bildschirmauflösung des normalen QLs. Mittlerweile dürfte ein Großteil der QL-User über höhere Auflösungen verfügen (QPC, QXL, ATARI mit QVME, AURORA) und schon ist ProWesS deutlich besser zu nutzen. Auf die vielen größeren Updates bei vielen anderen Produkten als auch anderen neuen Produkten möchte ich hier nicht im Detail eingehen, da sie nicht unbedingt für alle Leser interessant sind.

Zuletzt noch ein Aufruf an alle Leser: Artikel sind nach wie vor herzlich willkommen. An Themen gibt es eine große Auswahl: von Hardware-Tests über Software-Test bis hin zu Erfahrungsberichten mit Public-Domain Software oder Anleitungen zu PD-Programmen, die mit englischer Anleitung kommen. Es wird sicher alles von den Lesern gedankt.

Überrascht haben mich die vielfältigen positiven Reaktionen auf meinen Internet-Artikel; ich hatte mich eigentlich schon auf heftige Proteste eingestellt. Nun, gut zu wissen, daß nicht nur ich die Dinge so sehe...

Besonderer Dank gilt wie immer allen Autoren dieser Ausgabe und natürlich Dietrich Buders Korrektur-Lese-Künsten. Alle Artikel der letzten Ausgabe sind perfekt, nur sein eigener, den er gesetzt nicht mehr probe-gelesen hat, der ist nicht so wie er sein sollte. Dafür Entschuldigung und die Korrektur folgt sofort.

Auch dieses Jahr findet ein QL-Treffen in den USA statt. Ich war in den vergangenen beiden Jahren drüben und auch den Mitreisenden hat es sehr gut gefallen. Auch dieses Jahr werde ich mich wieder um gemeinsame, preiswerte Flüge und dergleichen bemühen, Interessenten sollten sich daher sobald wie möglich bei mir melden.

Bis zur nächsten Ausgabe verabschiedet sich

Jochen Merz

Kleine Korrektur

Leider hatte sich bei "Mandelbrotmenge" von Dietrich Buder in der letzten Ausgabe ein Fehler eingeschlichen. Der Cantorstaub wurde nicht so dargestellt wie er zu sein hat. Ich hatte mir den Text zwar ausgedruckt, aber leider mit einem anderen Druckertreiber als Dietrich. Daher ist die Grafik falsch. Hier nun die Korrektur:

Cantorstaub

Ein nicht sonderlich attraktives, aber mathematisch sehr wichtiges Fraktal ist der Cantorstaub, den der Gründer der Mengenlehre Georg Cantor im Jahr 1870 einführte. Die Konstruktionsregel lautet sehr einfach: Aus einer Linie das mittlere Drittel herausnehmen.

Iteration	Länge
0 -----	1
1 -----	1/3
2 --- ---	1/9
3 - - - -	1/27

Nach jeder Iteration verdoppelt sich die Zahl der Linien und die einzelnen Linienlängen gehen auf ein Drittel des vorherigen Werts zurück. Die Linienzahl wird somit nach unendlich vielen Iterationen unendlich und die Längen werden zu Null; daher der Name Staub.



Wofür braucht man das "Pointer Environment" oder "PE für Anfänger"

Ralf Tenbrink

Da nicht alle QL-Today Leser vorher auch Quasar gelesen haben wurde ich gebeten, auch die Teile 1 und 2 dieser Reihe von Ralf Tenbrink zu bringen. Er hat sie mir freundlicherweise zur Verfügung gestellt, also los geht's mit Teil 1:

Im letzten QUASAR *[liegt schon etwas zurück, nicht wundern - Editor]* gab es einen Artikel von Helmut Boll über seinen Verdruß mit Handbüchern und Anleitungen. Er hatte sich dabei auf einen vorherigen Artikel bezogen, in dem etwas ähnliches stand. Ich habe mich nun auch dazu entschlossen, etwas zu diesem Thema zu schreiben. Dabei möchte ich die Handbücher und Anleitungen nur kurz erwähnen und dann sofort zu meinen

eigentlichen Artikel kommen.

Herr Boll hatte sich Gedanken darüber gemacht, daß viele Anleitungen in englisch sind und daß viel Fach-Chinesisch in den Anleitungen steht. Leider ist das nicht nur so, wenn es um den QL geht, sondern man findet das auch in allen anderen Bereichen rund um den Computer. Da ich sehr viel mit dem PC und MS-DOS zu tun habe, kann ich nur versichern, daß es dort noch viel schlimmer ist. Man muß aber auch die QL Programmierer verstehen. Sie versuchen natürlich, ihre Produkte an einem großen Markt anzubieten. Dieser Markt heißt nun mal nicht Deutschland sondern eher "International". In England, Amerika und Europa gibt es wesentlich mehr Anwender von QL Software als in Deutschland. Da das Erfassen von Handbüchern fast genauso aufwendig ist wie das Erstellen der Software (wenn man es richtig macht), kann ich sehr gut verstehen, daß man das Handbuch in einer Sprache erstellt, die auch von den meisten Anwendern gelesen werden kann. Ich bin sehr froh, daß Ergon Development aus Italien die Handbücher ihrer Software in Englisch verfasst und nicht in Italienisch.

Sicher ist die Verständlichkeit ein zweites, größeres Problem der Handbücher. Auch hier gibt es in der MS-DOS Welt viele abschreckende Beispiele. Handbücher von über 1000 Seiten und dazu "Kurzanleitungen" von einigen hundert Seiten sind keine Seltenheit. Aber leider sind dann immer noch viele Dinge ungeklärt, weil man einfach viel zulange braucht, um etwas zu finden. Leider gibt es auch kein Rezept für ein gutes Handbuch. Selbst die großen Firmen haben noch keins gefunden. Und die haben Leute, die nichts anderes machen als Handbücher zu schreiben. Glücklicherweise ist die QL Welt recht klein und man kann noch ohne Probleme den Händler oder die Programmierer neuerer Software anrufen und Probleme klären. Auch per DFÜ kann man die meisten erreichen. Ich habe mal versucht bei der Microsoft Hotline anzurufen. Für die Telefonkosten, in der ich auf Warteschleife hing oder weiterverbunden wurde, hätte ich das Programm fast neukaufen können. Bei einigen anderen Firmen kosten die Hotline sogar Geld *[ja, das nimmt mehr und mehr zu - meine Empfehlung: vor dem Kauf von Geräten (Drucker, Modems usw.) fragen ob die Hotline über 0190-Nummern zu erreichen ist. Falls ja, anderes Produkt kaufen! - Editor]*.

Für mich war es immer am besten eine Software erst einmal zu probieren und "rum zu spielen". Damit habe ich es geschafft, 90% aller Programme zu bedienen, ohne einmal ins Handbuch zu schauen. Selbst bei umfangreichen Programmen kommt man durch Probieren weit genug, um das



Konzept oder die Idee des Programmes zu begreifen. Anschließend wird einem das Handbuch viel verständlicher.

Um nochmal kurz auf die fremdspachige Anleitungen zu kommen, gibt es bestimmt innerhalb des Clubs die Möglichkeit, viel gefragte Handbücher zu übersetzen. Man müßte nur mal wissen, bei welchen Handbüchern sich der doch erheblich Aufwand lohnt. Ich bin gerne bereit, einige Dinge zu übersetzen, die von vielen gebraucht werden.

Nun aber zu meinem eigentlichen Thema. Ich war doch sehr überrascht, daß Herr Boll einige Dinge einfach nicht kannte (Things, Buttons, Minerva). An einer Stelle meinte er sogar, daß er das Pointer Environment noch nie vermißt hat. Wer das Pointer Environment nicht vermißt, hat den QL noch nie richtig benutzt (Natürlich nur meine Meinung.) Deshalb möchte ich auch jetzt mal etwas zum Pointer Environment schreiben, das für die Anwender gedacht ist, die noch keins haben, oder die sich schon immer gefragt haben, ob sie das Pointer Environment brauchen. Alle anderen können nun getrost den Rest des Artikel überspringen oder aber meine Fehler korrigieren. Im folgenden werde ich das Pointer Environment nur noch PE nennen, das schreibt sich schneller. Ich selbst kenne das PE auch nur von der Anwendung her. Das, was innerhalb des QLs vorgeht oder wie man das PE vom Assembler aus programmiert, will ich auch gar nicht wissen. Ich werde also mit Absicht den Artikel so einfach wie möglich halten.

Für mich ist das Wichtigste am PE die Möglichkeit des vernünftigen Multitasking. Multitasking bedeutet, daß verschiedene Programme gleichzeitig im QL arbeiten. Das geht natürlich nur mit genügend Speicher. Ich gehen davon aus, daß jeder, der heute noch mit dem QL arbeitet, eine Speichererweiterung und das TK2 Toolkit hat. Auch ohne das PE kann man schon mit EXQUILL und EX ARCHIVE beide Programme in den Speicher laden und anschließend mit CTRL-C zwischen den Programm und SuperBasic wechseln, wenn man mit Tricks sicherstellt, daß sie nicht den gesamten Speicher in Beschlag nehmen. Aber man merkt sofort die Probleme. Jedes neue Programm zerstört den Bildschirm des anderen. Die Programme benutzen alle den Bildschirm und man kann nicht mehr erkennen, was zu welchen Programm gehört. Das passiert auch wenn man nicht die PSION Programme nimmt. Die einzelnen Programme zerstören den Bildschirm des anderen. Das ist eine Schwachstelle am QDOS System, die das sinnvolle Multitasking fast verhindert.

Einige andere Schwachstellen des Systems wurden schon sehr früh vom TK2 Toolkit behoben.

Deshalb hat sich das TK2 auch zu einem Standard entwickelt. Wer glaubt auf das TK2 verzichten zu können, kann auf den QL verzichten. Auch das PE beseitigt Schwachstellen im QDOS und erweitert seinen Umfang um wichtige Funktionen. Es scheint sich ebenfalls zu einem Standard zu entwickeln, wie das TK2. Ich glaube, das hat es auch verdient. Vor allem in Deutschland erfreut es sich hoher Beliebtheit.

Das PE besteht aus zwei einzelnen Programmen. Zuerst lädt man PTR_GEN und anschließend WMAN. So wie der Befehl TK2_EXT neue Dinge zum QDOS hinzufügt, wird durch das Laden der beiden Dateien das QDOS System verändert. Nun nochmal zu unserem ersten Beispiel. Ich lade wiederum QUILL und ARCHIVE in den Speicher und schalte mit CTRL-C zwischen den Programmen hin und her. Jetzt wird ohne mein Zutun automatisch der Bildschirm gesichert und wieder hergestellt. Kein Programm zerstört den Bildschirm des anderen. Auch wenn ein Programm nicht aktiv ist, (d.h es ist zur Zeit nicht auf den Bildschirm sichtbar) kann es weiterarbeiten. Ich kann also ein langes Dokument mit QUILL ausdrucken und trotzdem mit ARCHIVE arbeiten. Das ist natürlich nur ein kleiner Teil, den das PE nun ermöglicht, aber für ein effektives Arbeiten einer der wichtigsten. Nun will ich aber auch gleich eine Schwachstelle vom PE erwähnen, die mich doch etwas stört. Da viele Programme nicht für das PE geschrieben wurden und immer noch Programme ohne Rücksicht auf das PE geschrieben werden, gibt es Programme, die nicht weiterarbeiten, wenn sie nicht aktiv sind. Im Klartext bedeutet das folgendes. Das Programm möchte etwas auf dem Bildschirm ausgeben. Da es aber von einem anderen Programm verdeckt ist, wartet es mit der Ausgabe bis es wieder vollen Zugriff auf den Bildschirm hat. Das kann unter Umständen das Multitasking erheblich stören. Besser wäre es, wenn das PE erlauben würde, daß auch inaktive Programme eine Ausgabe machen können. Mittlerweile gibt es auch für diese Schwachstelle Hilfsmittel, das zu umgehen, aber das würde nun den Rahmen dieses Artikels sprengen. Sicher kann mal jemand anderes über diese neuen Dinge (PEX, PIE usw.) berichten.

Aber nun zu den anderen wichtigen Dingen des PEs. Das PE kann man nicht einzeln kaufen. Es wird immer mitgeliefert, wenn man Programme erwirbt, die das PE brauchen. (QPAC2, QD, LineDesign und viele andere). All diese Programme haben einen ähnlichen Aufbau oder ein ähnliches Erscheinungsbild. Das PE liefert Möglichkeiten und Funktionen, die von anderen Programmen benutzt werden können. Für den Anwender hat das den entscheidenden Vorteil, daß die



Handhabung unterschiedlicher Programme sehr ähnlich ist. Jedes Programm hat Buttons, Scrollbars, Subwindows und Menu Items. Ja, ja, ich höre schon den Aufschrei wegen des Fach-Chinesischs, aber ich werde alle diese Dinge nun eingehend erklären.

Am besten werde ich das an einem Beispielpogramm erklären. Ich nehme dafür QPAC II. Dies ist ein Programm, um alltägliche Arbeiten mit Dateien zu vereinfachen. Also Löschen, Kopieren, Umbenennen oder Vergleichen. In diesem Programm sind fast alle Dinge, die auch in anderen Programmen für das PE enthalten sind. Unser Beispiel-Bild zeigt den Bildschirm eines QLs mit PE. Oben in der dritten Zeile des Bildschirms sieht man mehrere kleine Vierecke in denen Namen stehen. Das sind die sogenannten Buttons. Darunter sieht man den Bildschirm von QPAC II. Innerhalb des Fenster gibt es nun einen Befehlsbereich mit Buttons (die oberen zwei Reihen) und ein großes Arbeitsfenster.

In den oberen zwei Reihen des Fensters finden wir mehrere Buttons. Ein Button ist eine Befehls- oder Schaltfläche. Einige Buttons haben grafische Zeichen, andere Worte. Diese Buttons können nun mit der Maus, mit den Cursor-Tasten oder mit einer Taste der Tastatur angewählt werden. Gibt es einen einzelnen

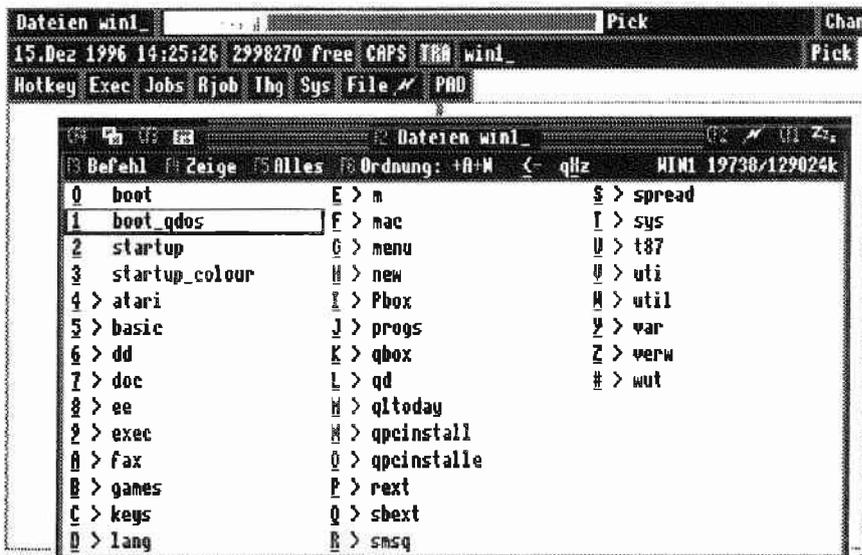
Tastendruck, so ist der betreffende Buchstabe unterstrichen. (z.B. das X in der zweiten Kopfzeile von QPAC II). Man aktiviert nun den Button "Befehl", in dem man mit der Maus auf das Wort "Befehl" fährt. Es erscheint dann ein Rahmen um das Wort "Befehl". (Man kann aber auch die Taste F3 drücken oder aber mit den Cursor-Tasten den Mauszeiger zu diesem Button fahren.) Mit einem Druck auf die linke Maustaste öffnet sich nun ein kleines Fenster. Das ist nun ein Menü mit mehreren Möglichkeiten (Items).

In der oberen Zeile des Fensters sind vier Buttons mit grafischen Zeichen. Von links nach rechts sind das: der Verschiebe-Button, der Vergrößerungs-Button, der Aktualisierungs-Button und der Schlaf-Button. Mit dem ersten Button kann man das Fenster hin und her bewegen. Das ist ganz wichtig bei Systemen mit größeren Bildschirmauflösungen

(Atari, QXL, QVME). Mit dem zweiten kann man die Größe des Fensters verändern. Vor allem dann wichtig, wenn man das Programm mehrmals gestartet hat (um zum Beispiel zwei Disketten miteinander zu vergleichen). Der Aktualisierungs-Button erneuert das Arbeitsfenster. Das wird dann notwendig, wenn ich die Diskette gewechselt habe und das Programm mir nun die neuen Dateien anzeigen soll. Mit dem vierten Button, dem Schlaf-Button, schließe ich das Fenster ohne aber das Programm zu beenden. Nun entsteht ein kleiner Button, der sich in die obere Reihe von Buttons einordnet (das ist der Button-Frame).

Innerhalb des Arbeitsfenster werden alle vorhandenen Dateien angezeigt. Mit der linken Maustaste kann ich nun mehrere Dateien markieren und mit der rechten Maustaste verschiedene Aktionen mit allen markierten Dateien ausführen (eine ausführliche Erklärung des Programmes würde den Rahmen dieses Artikels sprengen.) Ein weiterer

Teil des PE sind Pfeile rechts und links im Arbeitsfenster (erscheinen, wenn nicht alles ins Fenster paßt). Wenn ich mit der Maus auf diese Pfeile fahre, kann ich die Liste der Dateien bewegen. So kann ich mir alle weiteren



Dateien auf der Diskette oder Festplatte anzeigen lassen. Mit der linken Maustaste wird eine Spalte weiter gerollt, mit der rechten Maustaste eine ganze Bildschirmseite.

An der unteren Seite des Fenster ist der sogenannte Scrollbar oder Rollbalken. Mit diesem Balken kann ich mich schnell innerhalb einer sehr langen Dateiliste bewegen. Ich klicke nur auf das Ende des Balken und schon bin ich am Ende der Liste. Das geht natürlich auch, wenn ich auf den Anfangsbereich des Rollbalkens klicke.

Wie schon erwähnt, haben fast alle PE Programme den gleichen Aufbau. Wenn ich also nun den QD starte (das ist ein Editor für das PE) sehe ich ebenfalls meine gleichen Buttons wieder. Auch hier gibt es Verschiebe- und Vergrößerungs-Buttons, es gibt Befehls-Buttons und einen Rollbalken (Scrollbar). Dadurch wird natürlich die Handhabung der



Programme sehr vereinfacht. Viele Dinge sind direkt klar, ohne das man erst im Handbuch lesen muß. Und man muß sich auch nicht immer merken, mit welcher Taste in verschiedenen Programmen das gleiche erreicht wird.

Ich arbeite nun schon viele Jahre mit dem PE. Wohl schon von Anfang an. Auch heute noch wird das PE immer weiter verbessert und erweitert. Tony Tebby (der Programmierer des QDOS Systems) hat das PE entwickelt und er ist heute immer noch dabei. Viele neue Dinge sind zum PE hinzu gekommen und mittlerweile spricht man vom "Extended Environment" also der Erweiterten Benutzerumgebung oder kurz EE. Dazu gehört der Button-Frame, das Hotkey-System und auch wohl das Menü-System.

Wenn man QPAC II kauft, befindet sich zusätzlich zu den Informationen für die Handhabung auch Information zum PE. Man findet zuerst ein Tutorial (also eine Übungsanleitung) zum PE und den neuen Konzepten. Auf der Diskette befinden sich auch Übungsprogramme, um sich mit den neuen Dingen vertraut zu machen. Es gibt sogar eine deutschsprachige Version von QPAC und dem Handbuch.

Ich werde in einem nächsten Artikel gerne noch etwas über diese neuen Dinge des PE schreiben. Ich hoffe, daß ich die meisten Leser nicht mit zu einfachen Dingen gelangweilt habe. Ich würde mich freuen, mal eure Meinung zu hören, damit ich weiß, ob ich so weiter machen soll. Sagt der Redaktion Bescheid, die kann mich über das MAUS-Netz per DFÜ am besten erreichen. Natürlich freue ich mich auch über einen Anruf oder Brief.

■

BASICV - Teil 2

H.P. Recktenwald

Kommen wir zum zweiten Teil, dem Assembler-Listing:

Anmerkungen und Schreibweise:

** ENTRY" listet die Registerverwendung beim Einsprung auf,

** EXIT" dasselbe beim Verlassen eines Programmteils, nicht aufgeführte Register bleiben unverändert,

"xx" steht bei ungewissem Inhalt.

">" in der ersten Kommentarspalte deutet auf einen Subroutinenaufruf,

"?" steht bei bedingten und

"." bei unbedingten Sprüngen.

":=" drückt eine Wertezuweisung aus, "!=" heißt "ungleich".

"dbf" erscheint in der Schreibweise "dbra".

(adr) steht für den Speicherinhalt an der Stelle adr.

Endungen "a", "i" und "m" an den Memonics werden nicht geschrieben.

↩

MAUS an QXL2

Carlos Brennecke

Als ich meinen PC bekam, hatte der Händler Windows installiert, und die Maus lief einwandfrei. Aber im ersten Eifer hatte ich die Disk, die zur Maus gehört, sofort eingesetzt. Der zusätzliche Treiber tat das richtige, aber nicht unbedingt Nötige, denn auch ohne diesen läuft die Maus im PC! Nach Einsetzen der QXL2-Karte lief diese auch wunderbar! Man brauchte nicht einmal das lästige Drücken der Maustaste beim Einschalten wie bei manchen SER-Mäusen *[nicht alle Mäuse haben einen Schalter]*! Nach Aufmotzen und Neuformatieren des PCs lief die Maus prima, jedoch nicht mit der QXL2-Karte. Warum? Der Treiber war gelöscht worden! Nach neuem Installieren des Treibers war alles wieder in Ordnung. Beim Einsetzen von SMSQ/E war die Maus wieder tot! Sicherlich hat das nicht mit der Anwesenheit meiner Katzen zu tun, die immer dabei sind! Also Jochen fragen: Beim Konfigurieren von SMSQ/E muß man die Schnittstelle COM1 oder COM2, an der sich die Maus befindet, auf "none" stellen, sonst kommen sich beide Einstellungen in die Quere! Wo bei man beim Konfigurieren einiges beachten muß, bedingt durch beide Datei-Systeme. Die Datei SMSQ.EXE auf der Festplatte des PC ist nicht direkt mit Menuconfig zu bearbeiten, also muß man diese auf eine MSDOS-Disk kopieren mit COPY C:\QXL\SMSQ.EXE A:\SMSQ.EXE. Diese Disk ist mit Menuconfig unter SMSQ/E konfigurierbar. Jetzt zurück zu MSDOS. Die Disk mit der neu konfigurierten Datei mit der Anweisung COPY A:\SMSQ.EXE C:\QXL\SMSQ.EXE auf die Festplatte vom PC mit den neuen Einstellungen bringen. Jetzt kann man mit der Maus, dem Drucker, auf Deutsch (49) wie gewünscht arbeiten.

■

```

pagenen 20000
data 0
section bv

```

```

* SB-Funktion in das System einreihen

```

```

go
lea bastab(pc),a1
move.w #10,a2
jsr (a2)
moveq #0,d0
rts

```

```

bastab dc.w 0,0,1
dc.w basic-*
dc.b 6,'BASICV'
ds.w 0
dc.w 0

```

```

bsv_rbn moveq #-12,d0
bsv_rrt add.l $5c(a6),a0
move.l a0,$58(a6)
bsv_xit move.l $58(a6),a1
subq.l #2,a1
move.l a1,$58(a6)
move.w d0,0(a1,a6.1)
moveq #0,d0
moveq #3,d4
bsv_rbp rts

```

```

* entry to SB-Funktion

```

```

BASICV moveq #126,d1
move.w #11a,a1
jsr (a1)
moveq #0,d0
trap #1
move.l a0,a4
move.l d1,d5
moveq #0,d7
moveq #-15,d0
lea 8(a3),a2
cmp.l a2,a5
bos.s bsv_rbp
beq.s bsv_dfj
bsr fetchlv
bmi.s bsv_nst
move.l 0(a6,a1.1),d7
subq.w #1,d1
beq.s bsv_tjb
ber bsvjob
beq.s bsv_nst
bst.l tst.l d7
bpl.s bsv_chk

```

```

move.l d5,d7
bsv_chk bsr bsvckid
bsv_rrt add.l $5c(a6),a0
move.l a0,$58(a6)
move.l d0,d4
bmi.s bsv_xit
move.l a5,a2
bsr fetchst
bsv_rrt bne.s 0(a1,a6.1),d2
ble.s bsv_rbn
move.l a1,d6
move.l a0,a5
moveq #2,d0
move.l d7,d1
move.l d7,d2
trap #1
move.l a5,a0
move.l d0,d4
bne bsv_rrt
cmp.l d7,d5
beq.s bsv_nsp
move.l d7,-(sp)
move.w d3,-(sp)
pea retpri(pc)
moveq #11,d0
move.l d7,d1
moveq #0,d2
trap #1
bsv_nsp move.l d6,a1
move.l a5,a0
move.w sr,d0
trap #0
move.w d0,-(sp)
pea retusr(pc)
bsr idf.tst
add.l $5c(a6),a0
lea -6(a0),a1
move.l a1,$58(a6)
moveq #3,d3
and.b d5,d3
moveq #0,d1
tst.l d0
bmi.s bsv_err
move.l d6,a3
add.l d4,d6
move.l d6,a0
move.l d4,d6
bmi.s bsv_err
ror.b #1,d6
bmi.s bsv_err

```

```

try calling jobs variables
> job-id from qdos no.
clear parameter stack
? ret errnum
all (remaining) parameters
> vari name
? parameter error
? length error
save string ptr
old parameterspace
mt.jinf
job to test
top job in chain
check for valid id
? dest. is not a valid job
? don't halt the calling job
the dest.jobs id
and priority
care for return to re-activating
mt.prior
halt dest. job
string ptr back
bv-memory
supervisor
store old processor state
care for return to standard mode
> find variname, type in d5, jdt in a5
leave some space
clear remaining parameter space
type to return
prepare error/string return
? name error
a3:=dest.A6
vari_ptr
a0:=valueptr(a3)
? -ve ptr undef'd
? odd ptr not valid

```

```

add.l 40(a3),a0
cmp.w #0,0,d5
bos.s bsv_ndm
move.l (a0),a0
add.l 40(a3),a0
add.l a3,a0
bsv_ndm move.w (a0),d1
subq.b #1,d5
beq.s bsv_stg
subq.b #1,d5
beq.s bsv_fp
subq.b #1,d5
bne.s bsv_err
bsv_int addq.l #4,a1
move.l a1,$58(a6)
bra.s bsv_int

bsv_fp move.l 2(a0),d2
bsv_err move.l d2,2(a1,a6.l)
bsv_int move.w d1,0(a1,a6.l)
move.l d3,d4
moveq #0,d0
rts

* error return
bsv_err subq.b #1,d5
beq.s bsv_int
move.w #8000,d1
subq.b #2,d5
beq.s bsv_int
move.w #0F0D,d1
move.l #A60FC1B2,d2
bra.s bsv_fp

* d4:ptr to vari value d5.b type-1
* a1:r1-stack a0:ptr to vari(strg) a3:Job-A6 a5:dest.jdt
bsv_stg addq.l #4,a1
move.l a1,$58(a6)
move.w d1,0(a1,a6.l)
beq.s ptr_ret
* some qdos vectors mustn't be called while in supervisor mode
move.w 4(sp),sr
addq.l #1,d1
and.b #-2,d1
move.l d1,d7
move.w $11a,a1
jsr (a1)
move.l $58(a6),a1
sub.l d7,a1
move.l a1,$58(a6)
return

```

```

vari-address
type
? not dim'd
disp into vari memory
base disp
address of 1st dim'd item
len/int/exp
? string
? f.p.-number
? not an integer
- wert übergeben
<return f.p. result
<return integer result
integer or f.p.
ok
? empty string
? -0 = 32768 integer
-!e601
- f.p. error return

* re-fetch dest.job ptrs which might have been changed after bv.chrix
trap #0 halt again
move.l 88(a5),a3 dest.job-A6
move.l 40(a3),a0 and variable space
add.l d4,a0 ptr to the string
cmp.w #0300,d5 ? not a dim'd string
bos.s bsv_nds disp to dim'd values
move.l 0(a3,a0.l),a0 base ptr to values
add.l 40(a3),a0 address
bsv_nds add.l a3,a0
move.w (a0)+,d0 advance to chars and inst. count.w
move.w d0,0(a1,a6.l)
subq.w #1,d0
bsv_put move.b (a0)+,2(a1,a6.l)
addq.l #1,a1
dbra d0,bsv_put ? fetch string data
pru_ret moveq #1,d4 vari.type string
moveq #0,d0 ok
rts

* return procedure to resetting old processor state
retusr trap #0 supervisor to
move.w (sp)+,sr restore old processor state
rts

* return procedure to restoring old job priority
retpri moveq #11,d0 mt.prior
move.w (sp)+,d2 old priority
move.l (sp)+,d1 job
trap #1
rts

* fetch (an unlimited number of) parameters from basic
* long integer(s):
fetchlv move.w $118,a1 ca.gtlin
moveq #1,d1 flag: long integer
moveq #15,d4
and.b 1(a6,a3.l),d4 type
subq.b #1,d4
bne.s ftv_num ? maybe numeric
* string(s):
fetchst move.w $116,a1 ca.gtstr
moveq #0,d1 flag: string
moveq #0,d4 prepare error return ok
cmp.l a3,a2
b1s.s ftv_nix ? ret 0 if no parameter passed
move.l $58(a6),d3 bv.rip
sub.l $5c(a6),d3 bv.ribas
movem.l d1/d3/e5,-(sp) flag, size of basic-stack, top of parm.table(s)
move.l a2,a5 upper bound for this call
jsr (a1) , fetch value
move.l a5,a3 discard fetched parameter table(s)

```

```

movem.l (sp)+,d1/a0/a5 d1:flag a0:ri.stack a5:param.top
move.l d0,d4
bne.s ftv_nix
subq.w #1,d3
ftv_nix rts

*
* assembler routines accessible from any job and at any processor state
*
* find the (top) job the name of which
* was passed as .w-counted string at (a1,a6)+, starting ai=($58(a6))
* ENTRY 0(a1,a6)+:count.w&string a4:sysvari a6:own ref.addr
* EXIT 'eq':job not found 'mi':use current job d7:job-id d0-d4/d6/a1/a2:xx
bsvjob moveq #33,d3
moveq #0,d7
moveq #0,d0
move.l 104(a4),a2
move.w 98(a4),d1
move.w d1,d4
lsl.w #2,d4
lea 4(a2,d4,w),a2
move.w 0(a1,a6.l),d2
bmi.s bsj_rbn
beq.s bsj_xit
move.l a3,d6
bsj_jnm move.l -(a2),d4
bmi.s bsj_nxj
move.l d4,a3
ror.b #1,d4
bcs.s bsj_nxj
add.w #104+8,a3
cmp.w (a3)+,d2
bne.s bsj_nxj
move.l $58(a6),a1
move.w d2,d4
subq.w #1,d4
bsj_cmp addq.l #1,a1
move.b 1(a1,a6.l),d0
move.b (a3)+,d7
eor.b d7,d0
and.b d3,d0
dbne d4,bsj_cmp
bsj_nxj dbeq d1,bsj_jnm
move.l d6,a3
bne.s bsj_rbn
move.l (a2),a2
move.l 16(a2),d7
move.w d1,d7
moveq #1,d0
rts

```

```

bsj_rbn moveq #-12,d0 err.bn
moveq #-1,d7 'eq' on error
bsj_xit rot.l d7 'mi' for current job
rts

* check d7 for a valid job-id or find id of job number
* ENTRY d7 id/qdos-no. a4 sysvari
* EXIT d0 errnum d1 q.no. d7 job-id a2 jdt
bsvckid moveq #-2,d0 prepare err.nj
cmp.w 98(a4),d7 sv.jbmax
bhi.s bsc_ret ? job doesn't exist
moveq #0,d1
move.w d7,d1
cmp.l d7,d1 ? id fully specified
bne.s bsc_tid
move.l 104(a4),a2
lsl.w #2,d7
move.l 0(a2,d7,w),a2 jdt
move.l 16(a2),d7
move.w d1,d7
bsc_tid moveq #0,d0
bsc_ret rts

* this routine can be used by any program to find a basic variable ptr
*
* vari not found, build type conforming error return
idf_ndf moveq #1,d5 parameter type string
moveq #0,d1 len
move.w 0(a1,a6.l),d1 "%"-37 "$"-36
add.l a1,d1 last char
add.b 1(a6,d1.l),d2 -ve "$" 0 "%" +ve f.p.
bcs.s idf_nst ? undef'd string
sne d5
addq.b #3,d5 set to 0002 f.p 0003 integer
idf_nst rts

* ENTRY d7 owner/job-id of .w-counted variname at 0(a1,a6.l)+ a4 sysvari
* EXIT d0:errnum d1/a2/d3:xx d4:address
* d5:w:use|type d6:source/job-A6 a2:xx a5: dest. j.d.t.
idf_tst bar.s era_fnm
bne.s idf_ndf ? name not present, ret err.nf
moveq #12,d0 prepare err.bn
move.w #fff0f,d5
and.w -4(a2,d6.l),d5
beq.s idf_nix ? usage not defined
move.b d5,d1
beq.s idf_nix ? dim'd or string slice

```

```

move.l 0(a2,d6.l),d4
bpl.s idf_addr
not.l d4
beg.s idf_nix
not.l d4
lsr.b #1,d4
bos.s idf_nix
add.b d4,d4
idf_addr subq.b #4,d1
bos.s idf_var
subq.b #2,d1
bos.s idf_nix
subq.b #2,d1
bcc.s idf_nix
idf_var moveq #0,d0
idf_nix tst.l d0
rts

*
* this routine can be used by any program to find a basic name table
*
* find the .w-counted name at (a1,a6)+ owned by job with id in d7
* ENTRY a1(a6) string a4 sysvar1
* EXIT d0 ernum d6 dest.job-A6 -(a2,d6) name list in dest.job a5:jdt
m setstr d1/d2/d3/d4/d7/a1/a3/a4/a5/a6
a setstr a5
p setnum [ .instr(m,a)]/3*4 ; this way {p} safely points to stacked reg A5
era_fnm moven.l [m],-(sp)
move.l sp,a5
move.w sr,d0
trap #0
move.w d0,-(sp)
add.l a6,a1
move.l 104(a4),a2
ls1.w #2,d7
move.l 0(a2,d7.w),a2
move.l 88(a2),a6
move.w a6,d0
ror.b #1,d0
bos.s enm_nfd
* excessive odd-ness and sign testing done for safety if dest.job is not a
* basic-job, as there is no trivial means to checking for this kind of jobs.
era_objb move.w (a1)+,d0
ble.s enm_nfd
move.l 24(a6),a2
move.l 28(a6),d6
lsr.b #1,d6
bos.s enm_nfd
add.b d6,d6
restore ptr

? an address
? till unknown
? odd ptr
restore ptr
? 1/2/3 value
? 4/5 sbasic defn
? 8/9 not a loop index

```

```

sub.l a2,d6
ble.s enm_nfd
lea 2(a2,d6.l),a2
lsr.w #3,d6
add.l a6,a2
move.l a2,d4
bmi.s enm_nfd
ror.b #1,d4
bos.s enm_nfd
move.l a6,a5
add.l 32(a6),a5
move.l a5,d4
bmi.s enm_nfd
ror.b #1,d4
bos.s enm_nfd
moveq #-33,d4
enm_nxn subq.l #8,a2
move.l a5,a4
add.w (a2),a4
enm_nam cmp.b (a4)+,d0
bne.s enm_ntb
move.w d0,d1
subq.w #1,d1
move.l a1,a3
enm_cnr move.l (a4)+,d2
move.b (a3)+,d3
eor.b d2,d3
and.b d4,d3
enm_eqc dbne d1,enm_cnr
enm_ntb dbeq d6,enm_nxn
beq.s enm_oke
enm_nfd moveq #-7,d0
enm_oke sub.l a6,a2
addq.l #2,a2
exg a6,d6
move.w sr,d1
move.b d1,1(sp)
move.w (sp)+,sr
movem.l (sp)+,[m]
rts

? none or too many
variables searching top down for speed
count of sbasic names
top of name table
? -ve
? odd
address of name list
? -ve
? odd
mask to uppercase letters
ptr to names ptr
? other length
length count
name
string
? next char
? more names
? found
err.nf

```

Da das Listing ziemlich lang ist und wohl kaum abgetippt wird, jedoch vieles darin sehr interessant ist, wurde es platzsparend gedruckt. Der assemblierte Code ist in der JMS-Box zu finden und wird natürlich auch auf der Cover-Disk mit Heft 6 erscheinen!

LDump für den HP

Deskjet 500

Schönheit ist ein Abfallprodukt der Präzision

Helmut Friedrich

Freie Skalierbarkeit, Anwahl eines beliebigen Ausschnittes, millimetergenaues Positionieren auf dem Papier und verzerrungsfreier Ausdruck, machen einen Screendump erst wirklich sinnvoll. Die Möglichkeit, Screen's und Cut's direkt aus einer Datei auszudrucken, runden das Ganze noch ab.

Dies alles hat mich bewogen, LDump zu erwerben. Daß der Ausdruck außerdem auch noch gut, ja hervorragend aussieht, dies ist vom QL her wirklich ungewohnt. Was ich nun von dieser zwölf Jahre alten Schachtel zu sehen bekam, hat mich wirklich vor Begeisterung vom Stuhl gerissen, hin an's Telefon und dem Autoren meine Anerkennung ausgesprochen (mache ich sonst nie). Bislang habe ich QDesign's Laserdump für die High-End Lösung gehalten. Nun sehe ich, daß LDump wirklich alles bisher Dagewesene weit übertrifft. Jedoch ergänzen sich QDesign und LDump hervorragend.

Es war von den Entwicklern von SMSQ/E sehr weise, SDUMP_rext nicht fest mit einzubauen. LDump, mit ldp_dmo_byt, gehören inzwischen zum festen Bestandteil in meinem BOOT. Jedoch war der Anfang nicht gerade leicht.

Der Einstieg

Beim Booten fällt die Aufforderung auf: "BITTE ERST DAS HANDBUCH LESEN, sonst ist das Folgende zwecklos." Später sogar: "Nebenbei: Haben Sie das Handbuch gelesen?" Nun, ich habe es gelesen und es war zwecklos! Ich habe zunächst nur den Teil verstanden, der schon einmal im QUASAR veröffentlicht war und in dem verheissen wird was LDump so alles kann.

Die mir vorliegende Version war auf 75 dpi eingestellt, doch habe ich kein Interesse an einem 75dpi-Ausdruck! Ich ziehe den Draft- oder Tintensparmodus vor. Mich interessiert, wie ein Ausdruck in 300 dpi aussieht. Darum versuchte ich den entsprechenden Datensatz zu laden. Jeder Ladeversuch mit

```
LDP_DTL 'f1p1_HPM4_d300_LDT01'
```

führte zur Meldung: "nicht gefunden". Ich habe sehr lange gebraucht heraus zu bekommen, daß nur

```
LDP_DTL 'f1p1_HPM4_d300_'
```

zum Erfolg führt.

LDP_DMO

Zu einem unverzichtbarem Hilfsmittel wurde für mich das Programm ldp_dmo. Es benötigt eben auch jenes Ldp_dmo_byt. Anfangs hatte ich beides weggelassen und mir dadurch das Leben schwer gemacht. Nachdem ich es von den für mich störenden Eingriffen befreit habe, dazu gehören das Setzen der Baudrate und Reduzieren der Pufferung, habe ich dieses auf Hotkey gelegt:

```
ert hot_thing('/', 'sbasic'; 'lrun ldp_dmo', 'LDump')
```

Nach dem Start von ldp_dmo baut sich ein eindrucksvolles Bild auf: Rechts die "Zielscheibe", und im Fenster #2 werden zunächst sämtliche enthaltenen Prozeduren aufgelistet mit Zeilennummern und mit kurzen Angaben, was diese bewirken. Zwei davon sind für uns hier von Bedeutung: VARI und BILD. VARI liegt auch gesondert als "ldpvari_prg" bei, es enthält alle vorkommenden Variablen und Parameterbezeichnungen und trägt sehr zum Verständnis des Handbuchs und von LDump bei. Zur Arbeitserleichterung habe ich mir diese Prozedur ausgedruckt. Bei dieser Gelegenheit möchte ich auf die Seite 25 des Handbuchs hinweisen. Hinter allen dort aufgeführten Variablen gehört das %-Zeichen. Das Fehlen hatte zu schweren Mißverständnissen geführt. Es ist nun mal ein gravierender Unterschied (z. B.) zwischen mml% und mml! Das Gleiche gilt natürlich auch für die Variablenbezeichnungen auf dem Bildschirm. Nach diesem Ablauf werden nun im selben Fenster alle wichtigen Werte angezeigt. Dieses "Gesamtbild" zeigt also an, wie LDump im Augenblick eingestellt ist, und ich erkenne ob diese Einstellungen für mich brauchbar sind.

Da ist zunächst lil, genauer gesagt lil%, und lil% ist der Parameter-Nr. 13. lil% beinhaltet (z. B.) den Wert 132. Da ich nun einen Screen aus einer Datei ausdrucken möchte, gebe ich ein:

```
PRINT LDP_PRM(1il%,128) oder
```

```
PRINT LDP_PRM(13,128) (wegen der für mich besseren Übersicht ziehe ich PRINT dem DROP vor). 128 ist ein 1/4 der Screenbreite. Meine Unkenntnis hierüber hatte zur Folge, daß Schrott ausgedruckt wurde, und ich hatte den Eindruck, LDUMP hätte einen groben Fehler. Für Cut-Ausdrucke ist diese Einstellung nicht notwendig. Achtung: Nach WDUMP verstellt sich der Wert wieder auf 132! Genaue Berechnung siehe Handbuch S. 12.
```

Entsprechend nehme ich die weiteren Einstellungen vor:

```
PRINT LDP_PRM(mml%,0):rem abstand von links 0 mm
```



PRINT LDP_PRM(mmb%,86):rem bildbreite 86 mm

PRINT LDP_PRM(mmo%,0):rem abstand von oben 0 mm

Für häufig wechselnde Einstellungen habe ich mir folgenden Text zurecht gelegt. Er läßt sich aus QD mit SBAS/QD starten und mit ALT Shift Space zeilenweise zurückholen:

```
HOT_STUFF"LDUMP#'flp2_irgendwas_SCR"  
HOT_STUFF'PRINT LDP_PRM(0,0): rem  
abstand von links in mm'  
HOT_STUFF'PRINT LDP_PRM(1,86): rem  
bildbreite in mm'  
HOT_STUFF'PRINT LDP_PRM(2,0): rem  
abstand von oben in mm'  
HOT_STUFF'PRINT LDP_PRM(13,128): rem lil  
einstellen'
```

ink: -1 lasse ich so stehen. Nun jedoch zu xco:0! Diese Einstellung steht auf photographisch negativ und für die Positivwiedergabe setze ich hier: PRINT LDP_PRM(xco%,-1). Auch in diesem Punkt ist das Handbuch völlig unklar!

mde (mde%) steht auf 0, für Mode 4.

pxb (pxb%) stand bei mir auf 640 und pxh (pxh%) auf 480. Die Einstellungen für den normalen QL-Screen von LDP_PRM 3 - 6 sind natürlich 512 und 256. Siehe auch Seite 25 des Handbuches.

lbt:48 (lbt%) muss so bleiben. ngv:0 (ngv%) sollte auch so bleiben, ngv:-1 kehrt lediglich Schwarz in Weiss um und erzielt nicht den gewünschten Effekt wie xco; dies ist ebenfalls im Handbuch sehr unklar!

Nun etwas zur LDPSHD Rasterung: shd2% und shd4% haben den gleichen Inhalt mit dezimal 21845 oder binär 01010101010101. Dies bewirkt einen nur zarten Unterschied in der Schattierung zwischen rot und grün. Mit

```
PRINT LDP_PRM(shd4%,30447)
```

läßt sich ein deutlicherer Kontrast erreichen. Die übrigen Werte brauchen nicht geändert werden. In der rechten oberen Ecke des Fensters #1 fällt ein fettes f:2 auf. Es handelt sich um den Parameter shf%= 28 und wird im Handbuch auf Seite 35 genauer beschrieben. Er bewirkt eine "schärfere" Einstellung. Ich habe ihn mit

```
PRINT LDP_PRM(28,0)
```

auf 0 eingestellt. Dies bewirkt, daß nicht focussiert wird. Man sieht, der Autor hat wirklich an alles gedacht.

Nun zu Prae\$ und Post\$. Sie heißen in Wirklichkeit Prae% (=86) und Post% (=87), und es handelt sich natürlich um Prae- und Postambel. Lästigerweise ist jeweils ein Reset vorgesehen, erkennbar an dem großen E. Dies will ich natürlich so nicht haben, darum schreibe ich

zunächst: Prae\$ = Es\$&'*t300' und Post\$ = Es\$&'*rB'. In vari abgelesen: Es\$ = CHR\$(27). Gemäß Handbuch Seite 37 lassen sich nun durch Eingabe von LDP_MDE prae\$,post\$ Prä- und Postambel ändern. Dies geht natürlich auch mit:

```
PRINT LDP_PRM(prae%,Es$&'*t300) und  
PRINT LDP_PRM(post%,Es$&',*rB')
```

HP hat einen Tintensparmodus vorgesehen, den ich noch hinzufügen möchte. Doch zunächst rufe ich 'BILD' auf, um im Fenster #2 zu sehen, ob bis hierher alle Werte übernommen wurden. Dann speichere ich diesen Datensatz mit

```
LDP_DTS 'FLP1_hoch'
```

ab. Es wird beim Speichern eine Versionsnummer angefügt, die beim Laden weggelassen werden muß. Natürlich kann man nun auch den ganzen Code mit (z. B.)

```
LDP_SAVE 'flp1_HP500'
```

sichern.

Nun zum Tintensparmodus Druckerhandbuch Seite C-11 unten.

Für die Präambel: Prae\$&Es\$&'*r1Q' (schaltet Entwurf ein).

Für die Postambel: Post\$&Es\$&'*r2Q' (schaltet Entwurf ab).

Entsprechend habe ich nun eingegeben (mit FF nach Ausdruck):

```
LDP_MDE Prae$&Es$&'*r1Q', Post$&Es$&'*r2Q' &CHR$(12)
```

Das Ganze habe ich mit

```
LDP_DTS 'flp1_draft'
```

gesichert. Nun läßt sich LDump, je nach Bedarf, durch Laden des jeweiligen Datensatzes einstellen.

Ich hoffe, daß ich hiermit etwas zur Beseitigung von Mißverständnissen beitragen konnte. Für mich bedeutet LDump die "Software des Jahres". Wegen der Schönheit des Ausdruckes sehe ich auch gern über die beiden kleinen Mängel hinweg: Der untere Rand fehlt und der Ausdruck beginnt erst 3,5 mm unter dem von HP vorgesehenen Bereich. Auf jeden Fall kann ich LDump uneingeschränkt empfehlen. Ich packe nun auch wieder den Falkenberg-Scanner aus. Lange Zeit hatte ich ihn nicht mehr benutzt, denn was nutzen Bilder, die nur am Bildschirm schön aussehen?

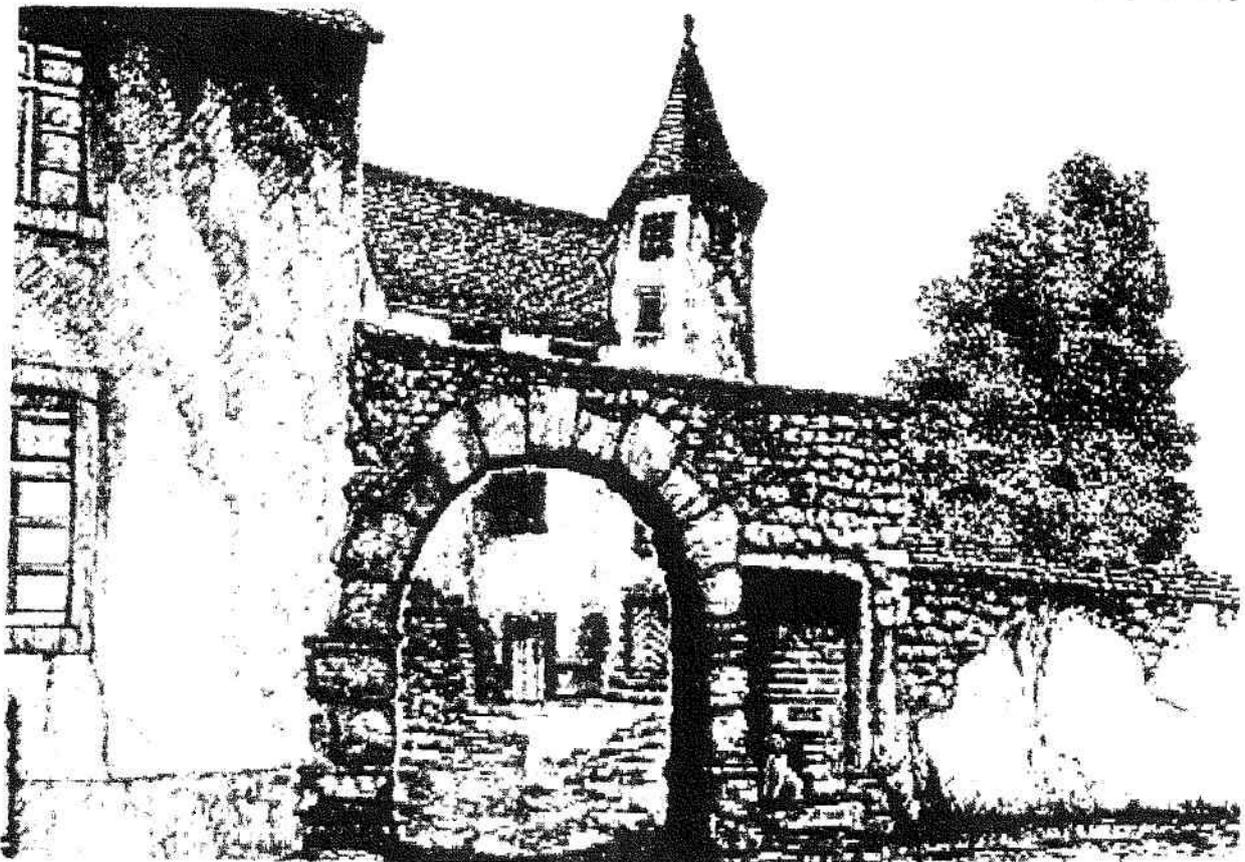
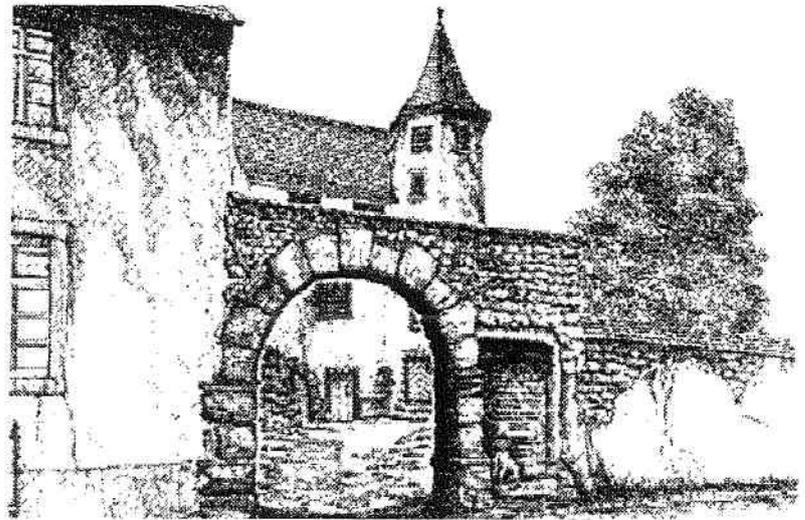
[Anmerkung des Editors: leider kann aufgrund des Aufbereitungsverfahrens die Ausdruckqualität der LDUMP-Bilder nicht annähernd so gut wiedergegeben werden wie Herr Friedrich sie uns zugesandt hat. Selbst bei kleinen Ausdrucken (beispielsweise 8x5 cm) sind die Ergebnisse umwerfend].

■



Titelbild in klein

**Noch ein schönes
Beispiel -
klein und groß**



Mein Bootfile

Thomas Menschel

Nachdem Jochen in der vorletzten Ausgabe 'sein' Bootfile für 4 verschiedene AtariQLs vorstellte, kann ich mir nicht verkneifen, die Sache für zwei schwarze QLs vorzustellen.

Da ist als erstes einer mit einer SuperGoldCard, Qimi-Maus und Falkenbergs Tastatur-Interface, beides im QL-Gehäuse. Naja, der Sockel für den IPC und damit für das Falkenbergsche mußte etwas 'verlängert' werden, eine kleine Stütze brauchte es auch und der Deckel rückte ein bißchen höher. Dann sind da noch zwei ED-Drives, ein Farbmonitor über RGB und ein Deskjet Drucker über ser1. Früher war da auch ein Modem dran, aber das macht inzwischen der PähZeh. Jaja, ich schäm mich ja schon, aber der kann einfach schneller und bunter; dafür hat er aber keine QXL, und wenigstens ein schwarzes Gehäuse. Der andere QL verfügt nur über eine GoldCard, die sich dafür aber schneller poken läßt. Hier ist ein Grau-Monitor dran, und der bunte FBAS-Ausgang hängt am anderen Eingang des Philips-Farbmonitors, der dafür vorne einen Schalter hat (und einen Fernsehempfänger für die Pausen). Hier sind zwei HD-Laufwerke dran. In beiden QL sind, umschaltbar, zwei Betriebssysteme verfügbar. Einmal Minerva und MGG (leicht umgemodelt), einmal Minerva und JS. Verbunden sind die beiden über so ein hauchdünnes Leitungchen mit popeligen Klinkensteckern in wackeligen Buchsen.

Damit ich nun nicht immer erst überlegen muß, was wo geladen ist, und was wo nicht geht, sind beide QL mit der gleichen Software bestückt. Folglich ist das Bootfile für beide gleich. Da ich auch mal die GC und die SGC austausche und die ROMs umschaltbar sind, muß alles beim 'anlassen' berücksichtigt werden. Bei Minerva sind mit PEX, PICE und Multibasic auch ganz andere Dinge möglich als bei MGG, bzw. JS. Dies bewerkstelligt ein Bootfile in 46 Zeilen. Also denn:

- 1 REMark boot
Einen Namen braucht das Kind.
- 2 v\$=VER\$: IF v\$='JSL1':m=1:sys=VER\$(-2):ELSE m=0:sys=163840
Ist das hier Minerva oder nicht? m als flag entsprechend setzen und nach sys die Basis der Systemvariablen
- 3 IF PEEK(sys+161)/16=2:sgc=1:ELSE sgc=0:POKE 114796,0
weil in sys+161 im 5.bit der Prozessortyp steht, und wir so erfahren ob es die SuperGoldCard oder 'nur' die GoldCard ist. sgc hält dies als flag für später fest, und wenn es die Goldcard ist sorgen wir gleich mal für etwas mehr Geschwindigkeit. Grins!
- 4 TK2_EXT:DNEXT bd:STAT flp1_:STAT flp2_:STAT ram1_:STAT ram2_:STAT ram3_:LRESPR
lng
TK2_EXT ist ja wohl jedem klar, in flp1_bd_ stehen alle zu ladenden Dinge. Die vielen STATs sorgen für später für weniger "heap fragmentation", ist aus der 640K Zeit übriggeblieben. Und das erste LRESPR lädt Lightning, weil dies wirklich als erstes sein muß.
- 5 LRESPR ptr:LRESPR wma:LRESPR hot:LRESPR his:LRESPR qp2:LRESPR men:LRESPR io2
Dies sind Pointer-Interface, WindowManager, Hotkey, die History (mein Vorschlag für den "Goldenen Löwen der Erweiterungen"), QPAC2, Menuerweiterung und IO2 (Vorschlag für Silbernen Bambi)
- 6 LRESPR qlb:LRESPR xtr:LRESPR e18:LRESPR cdi:ta=LRESPR(260):LBYTES traa,ta:TRA ta
Jetzt kommt noch QLIB_EXT, xtra ist ein kleines Extratool, e18 ein praktischer Editor, cdi ist CDISK und neu (vielleicht hat sich ja der Autor hier auch gemeldet) und dann noch TRA richtig setzen. Die Tra-Adresse in ta merken wir uns für weiter unten. Die Reihenfolge ist (außer Lightning) völlig egal
- 7 AS_LNK:AS_GO:FLP_SEC 2:PROT_DATE 1:_LNGON:_LNGZIP 2:DROP DMODE (0,0)
Die ersten beiden sind IO2-spezifisch. FLP_SEC 2 soll die Diskzugriffe sichern, ist aber genauso zweifelhaft wie PROT_DATE 1. Beides macht im entscheidenden Moment sowieso was es will. Lightning an schalten und dann das Scrollen auf 2 Zeilen setzen. Mehr bringt nichts, weil man nicht mehr mitlesen kann. DROP verwirft den Rückkehrparameter einer Funktion, genauso wie ERT, DMODE (0,0) setzt den Bildschirm auf Monitor-Mode und Mode 0
- 8 DROP W_OUTL(#0,510,256,0,0): QLIB_USE flp1_zub_,flp1_zub_,74,50,'0001011000'
Outline für Kanal 0 und die Parameter für QLIB
- 9 IF m=1
Nur bei Minerva. Bei anderen geht es nicht oder ist witzlos, spare mir also den Speicher dafür.



```

10 LRESPR pex20:LRESPR german:PEX_INI:PEON:POKE\212,128:EX pice
    PEX für die Fenster im Hintergrund und deutsche Tastatur laden, PEX rein und an, die berüchtigte
    Tokensierung von Minerva gleich wieder aus und PICE für die sichtbaren Fenster starten.
11 ALTKEY 'b','ex flp1_bd_multib_exe,con,con','': ALTKEY 'B','rjob job$(current)',
    ''
    mit 'b' geht ein Multibasic an, und mit 'B' wieder aus
12 ALTKEY 'n','OPEN#2,con:PAPER#0,0:INK#0,4:PAPER#1,2:INK#1,7: PAPER#2,7:INK#2,2:
Window 510,206,0,0>window#2,510,206,0,0: window#0,510,50,0,206:cls:cls#0',''
    mit 'n' bekommt das Multibasic meine bezugten Fenster
13 ALTKEY 'N','OPEN#2,con:PAPER#0,0:INK#0,4:PAPER#1,2:INK#1,7:PAPER#2,7:INK#2,2:
WMON',''
    mit 'N' die Standard Fenster
14 END IF
    Nun gilt es wieder für alle Varianten
15 DROP HOT_RES ('q','qsnap'):DROP HOT_WAKE ('.','button_pick'): DROP HOT_WAKE
('-','button_schlaf'):ALTKEY 't','TRA '&ta,'
    QSNAP resident, den Button-Frame-Wecker für die Maus, und den "Ab in die Heia". 't' setzt nach
    TRA 0 wieder die richtige Tabelle, dafür haben wir uns oben die Adresse in ta gemerkt
16 ALTKEY 'w','window#0,510,50,0,206>window#1,510,206,0,0: window#2,510,206,0,0:
paper#0,0:paper#1,2:paper#2,7:ink#0,4: ink#1,7:ink#2,2:csiz#0,0,0:csiz#1,0,0:
csiz#2,0,0: cls:cls#0',''
    Falls mal die Fenster 'vers...' sein sollten, 'w' bringt mir meine bevorzugten wieder
17 ALTKEY 'W','paper#0,0:paper#1,2:paper#2,7:ink#0,4:ink#1,7: ink#2,2:csiz#0,0,0:
csiz#1,0,0:csiz#2,0,0:wmon','': ALTKEY 'M','ed02 ed03 ram1_code_byt -nolink',''
    mit 'W' die Standard Fenster, 'M' ist der Kommandostring für den QMAC, ed02 und ed03 sind von
    edi18 (siehe weiter oben und unten) eingerichtete Editordevices. Da wird direkt aus dem Editor
    assembliert und das list geht direkt in einem anderen Editor.
18 DROP HOT_PICK ('s',''):DROP HOT_PICK ('e','EDI_01'):DROP HOT_PICK ('a','EDI_02')
:DROP HOT_PICK ('1','EDI_03'): DROP HOT_PICK ('m','QMAC'):ALTKEY 'j','cls#0:
xxx$=job$(current) :if len(xxx$)=0:print "System Job":else print xxx$,'
    Picks auf das StandardBasic, auf die drei edi18 jobs und den QMAC. 'j' schreibt in jedem Basic den
    Jobnamen auf den Bildschirm, manchmal weiß man nämlich nicht mehr wo man gerade ist
19 ALTKEY "c","CLS#0:FOR ppp=0 TO CH_HI(0):IF IS_OPEN(ppp):PRINT ppp;:IF TO_CHAN
(ppp)>=0:PRINT TO 3;TO_CHAN(ppp);:END IF :PRINT TO 6;CH_CHAN$(ppp);TO 30; CH_OJOB
(ppp);TO 33;:ooo=CH_OWNER(ppp): IF ooo:PRINT JOB$(ooo):ELSE PRINT 'System Job',''
    Damit zaubert mir 'c' eine Tabelle mit allen offenen Kanälen auf den Bildschirm. Ist doch erstaunlich
    was man alles in Hot_Keys machen kann.
20 DROP HOT_KEY ('y',CHR$(240)&'tw'):DROP HOT_KEY ('x',CHR$(27)&CHR$(242)&'0'):
ALTKEY 'z','ED JPEEK_W(158)','':ALTKEY 'p','ED LDEF('"'&CHR$(192)&CHR$(192):
ALTKEY 'f','cls#0:print "Noch ";free_mem/1024;" KB frei",'
    'y' und 'x' sind für text87, ersetzen dort das Durchhangeln um die Schrift zu ändern. 'z' bringt den TK2
    ED dorthin zurück wo man ihn verlassen hat, 'p' springt den ED auf der gewünschten Prozedur an, die
    2 CHR$(192) sorgen dafür das der Cursor nach dem Aufruf gleich zwischen den Anführungszeichen
    steht, 'f' zeigt den noch freien Speicher an
21 IF sgc=0:ALTKEY 'H','poke 114796,0:rem "HIGH SPEED",'':ALTKEY 'h','poke
114792,0:rem "LOW SPEED",'':ELSE ALTKEY 'H','cls#0: print#0,"SGC, geht nicht
schneller",'':ALTKEY 'h','cls#0: print#0,"SGC, geht nicht langsamer",'
    Unterschiedliche Hotkeys bei unterschiedlicher Hardware. Bei SGC gibt es nur eine Meldung, bei GC
    wird mit 'H' schnell und mit 'h' wieder langsam gepoked. Dafür haben wir uns oben das flag sgc
    gemerkt
22 ALTKEY "g","cls#0:DROP TICK(0,0):adr=ALCHP (32768):for n=1 to 5:M_M 131072,
adr,32768:end for n:RECHP adr:adr=TICK:if adr<10: print 'SGC 68020 24 MHz',:else IF
adr<13:PRINT 'GC 68000 24 MHz',:ELSE PRINT 'GC 68000 16MHz',:END IF :END IF ","
    Weil es keine Möglichkeit gibt den Schnell-Poke auch zu lesen, messe ich die Geschwindigkeit einfach
    mit einem Hotkey. TICK und M_M entstammen dem IO2. Da werden 32K Speicher reserviert, der
    Bildschirminhalt 5 mal dort hin gebracht und der Speicher wieder freigegeben. Die gemessene Zeit

```

unterscheidet ziemlich genau zwischen der langsamen und schnellen GoldCard, sowie der SuperGoldCard

23 HOT_GO:MEMV:EX EDITOR:EDI 'xxx':PAUSE 20:EDI 'asm':PAUSE 20: EDI 'list':PAUSE
20:EX c1mon:PAUSE 20: EX qmac:PAUSE 20: IF sgc:NET 1: ELSE NET 2

Hotkey an, MEMV zeigt am rechten Rand die Speicherbelegung und einiges mehr. Ein anderer Editor und die drei edi18 Jobs starten. Dazu den C1Monitor und den QMAC. Schließlich auf den unterschiedlichen Computern auch unterschiedliche Netzwerknummern. Die Pausen braucht er, sonst verschluckt er sich.

24 FSERVICE:EXEP 'dateien';'\IF1p1\NF1p1':EXEP 'dateien';'\IF1p2\NF1p2'
Netzwerkserver an und QPACs Dateifenster in den Buttonframe

25 BT_SLEEP 'pick':BT_SLEEP 'ljob':BT_SLEEP 'wake': BT_SLEEP 'sysdef'
QPACs pick und ljob sind manchmal ganz praktisch, wake um notfalls den Rest zu mobilisieren und sysdef braucht man unbedingt, weil QPACs Dateifenster intelligenterweise keine Subdirs anlegen können, und manchmal die Basics eben alle 'zu' sind

26 EX uhr;'7 482 0':HOT_DO 's':DUP:SPL_USE serlc:BAUD 19200

Eine kleine, schwarze Analoguhr am oberen rechten Rand, mit HOT_DO 's' (macht Pick auf StandardBasic) zurück ins richtige Fenster, DATAD\$ wieder zurück, Spooler auf das richtige Device und die Baudrate auf 19200, der Drucker kann das und antwortet sowieso nicht.

27 PAPER 2:INK 7:PAPER#2,7:INK#2,2:PAPER#0,0:INK#0,4: WINDOW 510,206,0,0: WINDOW#2,
510,206,0,0:WINDOW#0,510,50,0,206:CLS:CLS#0

Die Fenster richtig machen

28 HISTORY #0,25:HISTORY #1,10:HISTORY #2,20
SuperBasic eine History verpassen

29 IF m=1

und wenn es Minerva ist ein bisschen was sehr spezielles

30 MODE:EX flp1_bd_multib_exe,con,con:a=0:REPEAT b

einmal 'modieren' und den Multibasicjob starten. Variable auf Null und Schleifenstart

31 a=NXJOB(a,0):IF a=0:EXIT b

mit TK2s NXJOB durch die zahlreichen Jobs hangeln, wenn Null kommt wurde er nicht gefunden

32 IF 'sb.' INSTR JOB\$(a):EXIT b

Aha, da ist er ja

33 END REPEAT b

Erklärung nötig?

34 IF a>0

bei nicht gefunden können wir uns den Kram sparen

35 DROP HOT_PICK ('S','sb'):b=CH_FIND(a,-1,'CON_448x180a32x16|')

Erstmal einen Hot_pick auf den Multibasicjob einrichten und den Eingabekanal des selben ermitteln

36 DROP TO_QUEUE(b!0,'out1 #0,510,256,0,0'&CHR\$(10))

Dem Eingabekanal des Multibasic, aus dem im SuperBasic laufenden Bootfile heraus, ein paar wichtige Dinge verpassen. Als erstes eine Outline

37 DROP TO_QUEUE(b!0,'open#2,con'&CHR\$(10))

Dann Kanal 2 aufmachen

38 DROP TO_QUEUE(b!0,'paper#0,0:ink#0,4:paper#1,2:ink#1,7'&CHR\$(10))

Die Farben und

39 DROP TO_QUEUE(b!0,'paper#2,7:ink#2,2'&CHR\$(10))

die weiteren Farben,

40 DROP TO_QUEUE(b!0,'window#2,510,206,0,0:window#1,510,206,0,0:window#0,510,50,0,
206'&CHR\$(10))

und die Fenstermaße

41 DROP TO_QUEUE(b!0,'cls:cls#0'&CHR\$(10))

und durchlöschen

42 DROP TO_QUEUE(b!0,'history #0,25:history #1,10: history #2,20'&CHR\$(10))

History auch für den Multibasicjob

43 DROP TO_QUEUE(b!0,'cls#0:print "Multibasic "&job\$(current),'&CHR\$(10))

Ins Hauptfenster schreiben lassen, wer er ist



- 44 PAUSE 10:HOT_DO 's':PAUSE 10: HOT_DO 'S'
zurück ins SuperBasic und wieder ins Multibasic, das räumt ein bisschen auf.
- 45 END IF :END IF
die letzten IFs schliessen
- 46 PAUSE 10:NEW
Wenn es Minerva ist, löscht SuperBasic das abgelaufene Bootfile im Hintergrund, wenns nicht Minerva ist, sind wir sowieso nur hier

Tscha, das war's! 46 Zeilen, noch nicht mal übertoll. Das notwendige Rüstzeug für Basic, Assembler und Text. Die vielen ungewöhnlichen Befehle entstammen alle dem IO2-Toolkit. Ist in den Jahren gewachsen und läuft eindwandfrei.

Verschiedenes - Teil 2

Friedemann Oertel

Erfahrungen und Tips zu Festplatten: QUBIDE und QXL: - Für Festplattenneulinge -

Nachdem ich jahrelang beruflich recht zufrieden mit dem QL gearbeitet hatte, kam der Wunsch nach mehr Leistung. Minerva mußte her, Hermes und natürlich GC und SGC. Als Backupmedium verwendete ich zwei ED-Laufwerke, dann dazu ein 5 1/4"-Laufwerk.

Doch irgendwann begannen die Probleme, denn viele Megabytes an Daten, vor allem DOCs und QDesign-Pages mußten verwaltet werden, Backups waren auch notwendig. Alles sollte in einer vernünftigen Reihenfolge gespeichert und Upgedatet werden. Da waren die Floppies bald an der Grenze ihrer Möglichkeiten angelangt. Bei Davide Santachiara hörte ich, daß ein gutes, schnelles Festplatteninterface in Entwicklung war. Es sollte mit IDE-Platten, also AT-Busplatten nahezu jeder Größe arbeiten. Die wurden, im Gegensatz zu den MFM- oder RLL-Platten des JFC-Interfaces, immer schneller, leiser und vor allem billiger. Als in der Zeitschrift IQLR aus den USA ein begeisterter Testbericht über das Interface mit einer 420er Seagate-Platte zu lesen war, bestellte und erhielt ich gleich bei Ron Dunnet, einem sehr hilfsbereiten englischen Händler, eines der ersten Exemplare.

Damit begann für mich, einem absoluten Festplattenneuling, ungeahnter Ärger:

Die sehr sauber verarbeitete, kleine Platine wurde ausgepackt. Der Jumper für interne Spannungsversorgung war gleich gesetzt. Die EPROM-Meldung kam - alles bestens.

Beim Aufbau in meinem Tower fiel mir auf, daß trotz größerer Leitungslänge, die Goldcard mit der zweiten Konstellation in der Reihenfolge QL - QUBIDE - GC zuverlässiger startete:

Erste Konstellation:

GoldCard im oberen Slot des QL-Buses, QL ohne CPU im mittleren, Qubide darunter.

Zweite Konstellation:

Qubide steckt im oberen Slot des QL-Buses, darin die GoldCard. Im mittleren Slot der QL, der untere Slot ist frei.

Aber nun war die richtige Kartenadresse zu belegen. Zunächst ließ ich es bei der vorgeschlagenen Jumpe-

rung. Daß der Einsatz des Digitizers daran scheiterte, war mir damals nicht klar, da ich keine Ahnung hatte, was es mit der Adressierung und den Jumpers auf sich hat. Aus dem Handbuch konnte ich erkennen, welche Adressen von QL, Speicher-karte (GC, SGC, ...) und Betriebssystem (Minerva, MG, JS) zu verwenden waren. Ein Hinweis auf die nun nach Umjumperung trotzdem anschließbaren Geräte (Scanner, ...) wäre recht brauchbar gewesen.

Dann kam die erste Formatierung, - aus Basic heraus. Was war nun zu tun? Low- oder High-Level-Format? Was'n das??? Wann gehen welche Informationen verloren? Wieviel MB der Platte werden formatiert? Alle? Sektoren, nutzbarer Speicherplatz, Zylinder, Tracks, Woher weiß ich bei zwei gebrauchten 40MB-Western Digital-Platten, wieviele Tracks, Sektoren oder Zylinder sie haben, wenn diese Daten nicht vorlagen oder 'FAT wrong - reboot' gemeldet wurde? Und wie wird der Master-Slave-Betrieb zweier uralter WD-Platten, von der ich keine Daten mehr bekam, gejumpert? Da ist die zweite Platte die Tochterplatte der ersten. Die erste enthält alle wichtigen Informationen über die Drive-Map, also quasi das Inhaltsverzeichnis der ersten. Leute, steckt die Platten mal um, weil die erste die Map nicht mehr findet...! Aber nicht, weil diese Daten verloren waren, sondern weil es Timingprobleme beim Booten und Einlesen der Platten gibt! Wer dann noch glaubt, IDE ist IDE, der irrt. Nicht mal renommierte Hersteller halten sich an die IDE-Richtlinien. Man denke mal zurück an die Probleme vieler Floppies! Nun weiter: Die Slave-Platte fand, sie sei der Master, die wiederum war da anderer Meinung - und der Kopf knatterte am Anschlag. Ende einer Dienstreise. Doch bald kam ja die bestellte 420er Seagate, von der der amerikanische Autor in IQLR so geschwärmt hatte. Nun müßte ja alles klappen! Weit gefehlt, auch hier Ärger. Erst lief alles toll, ich eröffnete einige Unterverzeichnisse, fütterte die Platte mit jeder Menge Files und war hochofren über die unglaubliche Geschwindigkeit. Aber dann: Lange Wartezeit beim Initialisieren der Seagate, ... Machen wir es kurz: Die Platte wurde nach England geschickt und das EPROM von Phil Borman dem eigenartigen Timingverhalten der Platte beim Booten angepaßt.

Als das auch nicht alle Probleme beseitigte, tauschte ich die Seagate kostenlos gegen eine Conner um und siehe da: Alles lief wie geschmiert! Zumindest bis - aber das ist eine andere Geschichte. Der Witz an der Sache: Nun läuft auch die Seagate mit dem neuen ROM, laut Auskunft eines Bekannten sogar als Master-

Slave-Paar, was anfangs bei vielen Platten unterschiedlichen Typs/Marke Probleme bereitet hat - nicht nur am QL!

Eine Liste der definitiv funktionierenden Platte ist bei Ron Dunnet und auch in den JMS-Mailboxen zu erhalten.

Noch ein paar Zuckerl: Ich hatte natürlich zunächst keine Ahnung, daß man von Basic heraus erst nur einen Teil der Platte formatiert, und zwar maximal den, den QUBIDE verwalten kann (128 bzw. 256 MB). Den Rest erledigt dann das nun fertige Partitionsprogramm. Also z.B. 100 MB mit dem Basicbefehl formatieren, der Rest wird dann freigesetzt durch das Partitionsprogramm für z.B. 3 Partionen a je 100 MB.

Hier ein paar Beispiele für mögliche Partitionen meiner Platten:

Partitionierung einiger HDs: Western Digital, Conner

A - Partition auf WD AC2420H, 420 MB mit SGC

Drive 1	989 Tracks	15 Heads	56 Sectors		
Part. Nr.	NAME	START	END	SIZE MB	BLOCKS
1	WD-1	0	247	101	8
2	WD-2	248	495	101	8
3	WD-3	496	734	101	8
4	WD-4	744	989	100	8

B - Partition auf Conner CSF420A, 420 MB - 2/95 mit SGC

Drive 5	826 Tracks	16 Heads	63 Sectors		
Part. Nr.	NAME	START	END	SIZE MB	BLOCKS
1	CON-1	0	205	100	4
2	CON-2	206	411	100	4
3	CON-3	412	617	100	4
4	CON-4	622	823	100	4
5	CON-5	824	826	2	4

C - Partition auf WD 40 MB mit GC (Alte Kreissäge!)

Drive 1	782 Tracks	4 Heads	27 Sectors		
Part. Nr.	NAME	START	END	SIZE MB	BLOCKS
1	WD 42	0	782	40	4

D - Partition auf CONNER CSF425A MB mit SGC

Drive 7	839 Tracks	16 Heads	62 Sectors		
Part. Nr.	NAME	START	END	SIZE MB	BLOCKS
1	CON425-1	0	207	100	8
2	CON425-2	208	415	100	8
3	CON425-3	416	623	100	8
4	CON425-4	624	839	104	8

Nett auch, daß im Partitionsprogramm von Phil Borman Space (Leerzeichen) ODER NUN 'Y' die verlangte Bestätigung (confirm) ist, nicht das "O" (wie o.k.), was angegeben ist, sonst erhält man natürlich überhaupt keine bzw. die alte Partitionen, merkt es aber nicht unbedingt sofort!! Natürlich staunt man auch nicht schlecht, wenn man entnervt nach verzweifelter Booterei mal mit der dt. Tastatur und Minerva keine deutsche Tastenbelegung geladen hat, statt für Y(es) ein Z eingibt. Na ja... Wenn man mal gesehen hat, wie es geht und was am Ende passieren muß, ist es ja ganz leicht. Aber bis dahin!

Die wichtige Empfehlung von Ron Dunnet: Partitionen erst mal nur mit ca. 40 MB max. einrichten, sie sind dann ...

a) - leichter zu löschen und neu zu überspielen.

b) - auch vom (ehrwürdigen) QRAM und von QPAC2 (viel schneller) zu verwalten, bei 100 MB-Partitionen klappte bei mir manchmal das Anklicken der letzten 2000-3000 Files von gesamt über 10.000 bei gesamtem Baum bzw. Sortieren (letzte Files ansehen etc.) nicht mehr! Aber: Über Netz war das vom der 8MB-QXL noch möglich! Verrückt, was? Also: Kleine Subdirectories halten!

Spontan erschienen bei mir beim alten ROM V1.28 Fehler, wie Namen, die Files als auch Subdirectories darstellten, waren nicht mehr löscherbar, also diese Partition neu formatieren! Das ist nun vorbei. Auch gelöst: Gab man ein: Save 'WIN1_Grafik_', so war auf WIN1 im Subdirectory Grafik ein neues, unlöschbares Subdirectory entstanden. Ähnliches galt, wenn man eine Anzahl Files, ähnlich wie bei den Club-PD-Disketten, übersichtlicher innerhalb eines Unterverzeichnisses trennen wollte:

```
WIN1_GRAF_Linedesign
```

```
WIN1_GRAF_abc
```

```
...
```

```
...
```

```
WIN1_GRAF_zz
```

```
WIN1_GRAF_ _____ * <-- war problematisch!
```

```
WIN1_GRAF_
```

```
WIN1_GRAF_QDesign
```

```
...
```

```
...
```

```
WIN1_GRAF_xyz
```

```
WIN1_GRAF_xyz
```

* Problematisch, da: Beim Saven des vier Byte langen Mini-Basic-Programmes

```
'1:'
```

mit dem Befehl Save "WIN1_GRAF_ ___" war ein neues Unterverzeichnis entstanden. So kann man zwar schnell Files im Listing kommentieren, hatte aber ein neues Unterverzeichnis erstellt.

War auch gefährlich: Wenn man kein Basic-Programm geladen hatte und obigen Befehl eingab, sicherte man ein Programm mit der Länge 0. Raten Sie mal, was dann passiert? Man merkt es daran, daß das Netzwerk streikt, QPAC2 nicht mehr richtig arbeitet oder der WINEDIT(or) auf der Partition der Platte mehr Files meldet als er findet. Formatieren macht aber mit dem neuen Partitionsprogramm richtig Spaß, kann man doch einzelne Partitionen der Platte selektiv entfernen und wieder neu einrichten. Wie heißt es im ersten Handbuch: Sie werden Daten verlieren ...

Lange oder eigenartige Filenamen, die bei Netzwerkkopien mit einem vorgestellten N1_... zu lang werden und nicht mehr kopiert werden können und dann manche Backupprogramme (das alte Norback: QLib-Error) zum Aufhängen bringen. Macht nach ca. 25 erfolgreich kopierten ED-Disketten enorme Freude, wenn man nochmal neu beginnen kann, weil das Kopierprogramm aussteigt! Ich bin dann einfach auf QPAC2 umgestiegen, das bietet mehr Möglichkeiten, auch mit dem Betrachten und Vergleichen von Files. Manche Autoren (Molecular Graphics z.B.) lieben monströse Namen mit mehr als 30 Zeichen! (Abhilfe: Statt WCOPY ... to N1_ ... kann man mit NFS_USE



TMR, WIN1_ arbeiten.) Oder besser in SMSQE mit dem Befehl DEV_USE ... Empfehlung: In Norback sollte eine Option für den Umgang mit fehlerhaften Files/Disketten eingebaut sein, daß solche bei Auftauchen eines Fehlers in der Gesamtdatei ausgeblendet und nicht mit abgespeichert werden.

Hatte ich dann per Netzwerk eine Partition mit so ca. 100 Megabyte Files vom QUBIDE-QL auf die QXL.WIN der QXL-Karte kopiert, fand ich dann möglicherweise Einträge der Länge Null vor (siehe oben), oder die sonst wirklich hervorragende QXL-Karte meldet dann 'QXL at adress ... hex not responding' (meine Lieblingsmeldung) und schaltete unter Protest (= Verlust des RAM-Inhalts) zum DOS zurück. Denn: Gemeinerweise kopiert der QL erst einmal in die 8 MB RAM der QXL - und stürzt dann unter Datenverlust ab. (Anm: Dieses Problem trat bei neueren SMSQ/E-Versionen nicht mehr auf, da ich nun problemlos im Turbo-Modus arbeiten konnte.)

Abhilfe: Mit dem neuen Befehl WDDIR ein kleines Basic-Kopierprogramm schreiben - und dann per Multitasking dem Computer etwas anderes zu tun geben - dann wird der RAM-Inhalt der QXL geleert und gerettet, während der QL sich mit z.B. QPAC2 abmüht und keine Daten mehr über Netz sendet.

Zum Glück kann man nun im neuen Partitionsprogramm auch einzelne Partitionen löschen bzw. allein formatieren, ohne die anderen (WIN1_, WIN2_, ...) alle zu verlieren, das spart enorm Zeit und Nerven!

Mittlerweile verwende ich als Backup-Medium eine 2. und 3. WIN (a 420 MB), das macht das Leben einfacher und ist schon relativ preiswert. Man rechne sich mal den Diskettenpreis für 420 MB aus! Von der Arbeitszeit ganz zu schweigen, wenn man Stunden vor dem QL Diskjockey spielen kann. Noch besser ist allerdings ein Laufwerk mit Wechselmedien. Legt man die Partitionen WIN1 ... WIN4 und WIN5 ... WIN8 auf Platte 1 und 2, so kann man eine 3. Platte abgeschaltet lassen (Datensicherheit) und hat eine absolut geschützte Backupplatte! Einschränkung: Siehe oben: Datenverlust bei manchen Konfigurationen mit Nastas Multi-Master-Kabel, das 4 Masterplatten versorgt. Angeblich sollen Widerstände in den Daten- und Adressleitungen und kurze Kabellänge zur Platte diese Probleme beheben. Von Ron Dunnet empfohlen wird allerdings das oben beschriebene Master-Slave-Verfahren mit max. zwei Platten.

Das beiliegende Backupprogramm NORBACK sollte vielleicht noch zusätzlich einen konfigurierbaren BEEP haben, damit man während der endlosen Floppy-Kopiererei mal aus dem Zimmer gehen kann und zum Diskettenwechsel von seinem QL 'gerufen' wird. Trotzdem empfehle ich (Zeit und Nerven sparen !) dringend eine 2. Festplatte.

Mit dem Befehl ...

WIN_DRIVE (gewählte Partition), (PlatteNr),
(Partition der gewählten Platte)

kann man sich die gewünschten Platten bzw. Partitionen in beliebiger Konfiguration (!) einschalten. Mit WIN_DRIVE 5 dagegen (ohne weitere Parameter) wird die aktuelle Partition 5 abgeschaltet. Das ist beim Plattenwechsel nötig, sonst wundert man sich, warum z.B. das Syquest-Laufwerk die Platte nicht herausgibt.

Bsp.: WIN_DRIVE 1, 3, 2 - Die gültige, nun eingeschaltete Partition 1 (WIN1_, ähnlich wie FLP1_)

ist auf der physikalischen Platte Nr. 3 (Steckplatz Nr. 3) und wurde dort als 2. Partition dieser 3. Platte formatiert. Sicherheitshalber formatiert das neue Format- und Partitionsprogramm von Phil Bormann nun jede Partition auf Wunsch bis zu fünfmal !!!!!

Fehlermöglichkeit: Bitte warten, bis nach dem Befehl WIN_DRIVE der Cursor wieder auftaucht. Es gibt Ärger, wenn man zwischenzeitlich z.B. QPAC2 startet und die Map noch nicht gelesen ist. Noch'n Bug: Bei ROM 1.15 bewirkt die wiederholte Eingabe des Befehls WIN_DRIVE 1,1,1 das Abschalten (!) der Platte (was eigentlich WIN_DRIVE 1 tun sollte).

Aufhänger im alten SMSQ/E (aktuell: 2.77): Bei Eingabe von NET (ohne Nr.) erwartete der QL broadcasting und hing sich in SMSQ/E auf, das ist nun alles Geschichte. Das neue SMSQ/E für die Goldcard ist nicht nur aufgrund der hohen Geschwindigkeit sehr, sehr empfehlenswert! Leider traurig für mich: QDesign arbeitet nicht einwandfrei. Wer kann da helfen?

Aufhänger bei langen Files gab es bei meinem System zweimal: Einmal als ich bei fast vollem Speicher mit QD7 mein riesiges QTPI_log editieren wollte (man kann da so schön schnell und leicht die Steuerzeichen und unwichtigen Text entfernen) und bei der QXL-Card mit einem riesigen, 'zusammen-gemergeten' XChange_doc für die QUASAR-Artikel. Beim QTPI_log wurde der ganze Subdirectory-Inhalt auf die Länge 0 geschrumpft, weiteres Abspeichern oder Finden von Daten darinnen war nicht mehr möglich. Also wieder mal die Partition neu formatieren...

Eine absolute Intelligenzleistung meinerseits war folgender Test: Abschalten der 12V-Spannungsversorgung - die HD wird sooo schön leise wie bei meinem Laptop in Power-Save-Schaltung. Ob man aber, nachdem Daten auf die ruhende Platte oder die anlaufende Platte geschrieben wurden, diese wieder findet, ist zu bezweifeln. Witzigerweise hatte ich damit gerechnet und vor dem Härtetest meine Harddisk-Daten per Netz auf die QXL übertragen (dauerte ca. 14 Stunden!) und mit Erfolg getestet. Da aber meine Tochter ein verwirttes Progrämmchen von der Schule mitbrachte, ... Alles klar???? Also, bitte nicht nachahmen. Merke: Überlegen macht Überlegen.

Lädt man die Level 1 - Ramdisk von QRAMs alter RAMPRINT-Funktion, so überschreibt er den QUBIDE-RAMDISK-Treiber, der nicht ganz exakt dem von T. Tebby entsprechen soll. Probleme gibt es dann beim Kopieren in die RAMDISK, z.B. mit DiskMate 5 oder Cueshell (QLIB-Errors oder Startprobleme). Das läßt sich vermeiden, indem man entweder dieses RAMPRINT nicht lädt oder aber nach dem Laden den Befehl RAM_USE zzz o.ä. setzt, also die Ramdisk quasi umleitet.

Die Directory-Struktur der der DOS-Datei QXL.WIN unterscheidet sich erheblich von der des QUBIDE-Interfaces: Die QXL.win ist eine Datei auf der PC-Festplatte, die von Anfang an auf eine feste Länge formatiert wird. Hier erkennt man gut am Directory-Eintrag (0), ob dieses leer ist. Wurde einmal ein File darin abgespeichert, erkennt man in QPAC2 mit 'Zeige', daß Informationen hineingeschrieben wurden: Die Länge des Directory-Files ist nun nie mehr 0, die Namen der gespeicherten Files und weitere Informationen sind darin enthalten - oder sollten es zumindest sein. Somit erkennt man rasch leere Subdirectories (Länge 0). Löscht man alle Files aus dem Subdirectory,

bleibt allerdings die alte Länge. Das QUBIDE-Interface organisiert die Files anders. Subdirectories haben immer die Länge 0. Nun ist mir beim Kopieren per Netzwerk von QL nach QXL (altes ROM) passiert, daß Files erschienen, die unlösbar waren, gleichzeitig erschien bei weiteren Abspeicherversuchen bei zu 1/3 freier Platte die Meldung "OUT OF MEMORY". -> Neu formatieren - anfangs eine beliebte Übung. Schuld war das Verlangen, auch uralte Disketten auf Festplatte zu überspielen oder oben genannte Files der Länge 0.

Auch die Längenangabe der Files schien mit dem alten ROM 1.28 nicht ganz zu stimmen: Manchmal schrieb QUBIDE einfach Teile von Files in andere Files hinein. Dies könnte aber auch mit dem Prototyp des Multi-Master-Kabels zusammenhängen. Ich habe versucht, das mit einem eigenen Kopier-Programmchen zu umgehen und dabei auch noch einen File-Längencheck durchzuführen. Ich benutze den WDDIR- Befehl in den QUBIDE-TOOLS, der alle Files auflistet. Findet sich auf der PD-Disk von München.

Nicht erwähnt ist die nötige, enorme Anpassungsarbeit beim Umstieg auf Festplatte: Fast alle Floppy-Programme mußten umkonfiguriert werden. Mit DEV_Device kann man das recht elegant umgehen, ich mag das aber nicht so gerne, da mir das Warten beim Laden/Absaven von z.B. QUILL_docs aus großen Unterverzeichnissen viel zu lange dauert. Mittlerweile arbeiten fast alle meine Programme optimal, nur Text 87 und LDesign muß ich noch besser 'lernen'.

Arbeitet man mit der Goldcard mit nur 2 MB RAM, wird man feststellen, daß der Speicher sehr schnell schwindet, da die Directories bei voll genutzten 420er Platten enorm viel Platz brauchen. Abhilfe bietet das neue ROM (neuer als 1.30): Man kann hiermit mehrere Partitionen zu-/abschalten und damit sogar bei magerer RAM-Ausstattung Platte 1 / Partition 1 auf z.B. Platte 2, Partition 4 updaten.

Allerdings empfehle ich allen dringend, sich erst einmal kleine, überschaubare Partitionen und übersichtliche Subdirectories mit kurzen Namen zu schaffen und die zu kopierenden Disketten gründlich durchzuchecken. Nehmt erst eine kleine Partition, auf der Ihr erst mal ungetestete Files ausprobieren könnt! Fehler dieser Partition bzw. auf der wichtigen, ersten Partition wirken sich nach meiner Erfahrung nicht mehr so gravierend wie bei Version 1.28 auf die ganze Platte aus. Ich hatte keine Ahnung, wieviele Fehler auf Disketten ziemlich unbemerkt auftauchen können, die sich dann auf der Platte fröhlich ansammeln. Nicht nur bei, (einer ungenannten QL-Dame), wo plötzlich jede Menge unerklärlicher Lesefehler auftraten. Ursache: Eye-Shadow auf den Disketten! Man glaubt es kaum! Bis man darauf kommt Auch manche alte, mit TK3 abgesavte Files oder Disketten, die mit 84 Spuren formatiert wurden und auf verschiedenen Laufwerken, die nicht unbedingt für 84 Spuren geeignet waren, waren bei mir genauso eine Fehlerquelle wie Disketten von manchen exotischen Laufwerken. Das neue SMSQ/E berücksichtigt angeblich solche Probleme und TKIII gibt es bei R. Reköndt als upgedatete, kompatible Version. Viele solcher Fehler addieren sich halt bei 420 MB - das sind nun mal mindestens 150 ED-Disketten! Gute Dienste erwies mir da FPU, die Floppy-Disk-Utility von Ergon, mit der man viele Fehlerquellen von vorneherein erkennen und ausschließen kann. Auch

andere Diskeditoren sind hier sensibel, wenn's auch Zeit kostet.

Ärger machen auch allgemein Basicprogramme, die mit verschiedenen Versionen von QLib, Qsave oder verschiedenen Betriebssystemvarianten, also MG(G), Minerva oder SMSQ Q-gesavet wurden und nun mit neuen/anderen Versionen geladen werden. Also immer ein normal gesavtes Basic-File als Reserve halten. Aber letzteres hat mit QUBIDE wenig zu tun, eher mit den entsprechenden ROMs. Man merkt alles nur umso eher, wenn man sich immer das gleiche Boot-System zusammenstellt oder den QL mit den verschiedenen, umschaltbaren oder zuladbaren Betriebssystem-Varianten (JM, MG, Minerva, SMSQ(E)) fährt.

Trotzdem: Fazit aus der bisherigen Entwicklung von QUBIDE: Ein Beispiel gelungener Produktpflege. Das QUBIDE-Interface hinterläßt nun, ca. 1 Jahr nach seinem Erscheinen, mit der ROM-Version 1.41 einen ausgereiften Eindruck. Viele, vor allem moderne, preiswerte Platten laufen. Die Software wurde wesentlich verbessert. So kann man nun jede beliebige Partition auf jeder angeschlossenen Platte nach Belieben formatieren. Arbeitstempo und Datensicherheit sind selbst bei großen Datenmengen, hier 1.2 Gigabytes, sehr erfreulich.

Wie sagt Murphy? Fehler die möglich sind, passieren auch. So stand auch in einem Handbuch: Sie werden Fehler machen! Ich will ihm nicht widersprechen.

Ausblick: Man arbeitet zur Zeit mit dem Autor von Xover daran, mit dem QL und einer in gewöhnlichem DOS formatierten Syquest-Platte zu arbeiten. Das Basic-Formatieren und Partitionieren mit einem speziellen Programm entfällt dann. Das neue ROM V1.51 ist schon fertig. Man kann damit auch Fremdformate (<>QDOS) lesen/schreiben. Mit dem Befehl CSUM wird die MAP bei jedem Zugriff gecheckt. Dauert etwas länger, ist aber sicherer.

WIN_CSUM 1,1 schaltet die Überprüfung für Platte 1 ein, CSUM 1,0 wieder aus.

Ein Wehrmutstropfen: Auf meiner Conner-Harddisk lies sich zwar alles schön speichern, allerdings ging QPAC2 - und auch QRAM - Kopieren nicht mehr. Also 'Back to the roots': Wieder Downgraden zu Rom 1.41, da klappt alles. Noch' Ding: Bei der netterweise von Stuart in Solms kostenlos upgegradeten Gold-card (aktuelles ROM: V 2.49, INGOT 5, 25 ns schnell) MUSS (!!) unbedingt der Uhrumbau erfolgen, da das für andere Zwecke abgespeckte ROM auf die GC-Uhr zu- greift. Es kommt beim Booten des Systems keine Fehlermeldung 'Check the clock' mehr. Allerdings Initialisiert die Platte dann auch nicht mehr.

Wer sich eine Syquest-IDE-Wechselplatte EZ 135 (witzigerweise mit nur 128 MB) zugelegt hat, freut sich sicher über die sehr schnelle Platte, allerdings ist beachten, daß der Schreibschutz der Wechselmedien nur in MSDOS usw. funktioniert. (Frei nach Woody Allen: Was Sie schon immer über Festplatten wissen wollten, aber nie zu fragen wagten). Eine bedauerliche Nachricht: Syquest liefert keine EZ 135-Laufwerke mehr.

■

JOHANN MERZ SOFTWARE

Im stillen Winkel 12 • 47169 Duisburg • Germany
 ☎ 0203-502011 (Fax 0203-502012 Mailbox 0203-502013 & 502014)

QPC ist da! QPC, der brandneue QL-Emulator mit SMSQ/E ist nun fertig! Man kann jetzt SMSQ/E auf PC's und Laptops ohne extra Emulator-Hardware laufen lassen - wird in Software emuliert! 486 oder Pentium ist Voraussetzung. Der Preis ist DM 199,- für Kunden die SMSQ/E für andere Systeme bereits besitzen, andernfalls DM 249,-. Mit CueShell nur 40,- DM mehr!

**PREISSENKUNG: SuperGoldCard nur noch DM 499,-
 AURORA IST DA!!!**

**Ersatz für das QL-Mainboard und erweiterte Grafikauflösungen -
 Details in der nächsten QL Today - wer nicht warten kann: DM 359,-**

NEUHEIT: FiFi II - Zeigt nun die gefundenen Dateien an während weitergesucht wird! Zudem gibt es neue Optionen wie "nur nach Worten suchen", "keine Subdirectories durchsuchen", Dateien mit bestimmten Datei-Endungen durchsuchen oder nicht durchsuchen und einiges mehr. Mit deutscher Anleitung [V4.09] DM 54,90
 Upgrade von älteren Versionen DM 19,90

NEUHEIT: BASIC Linker - endlich ist modulares Programmieren in BASIC möglich! Mit deutscher Anleitung! Benötigt QD und QLiberator DM 49,90

QL Spiele

BlackKnight Schach DM 119,90
 Pipes DM 29,90
 BrainSmasher DM 39,90
 Arcanoid DM 39,90
 Firebirds DM 39,90
 QShang DM 39,90
 Diamonds DM 39,90
 The Oracle DM 39,90
 MineField DM 39,90
 Double Block DM 39,90
 The Lonely Joker 2 . DM 59,-
 SuperGamesPack ... DM 90,-

QL Ersatzteile

ZX8301 DM 24,90
 ZX8302 DM 19,90
 Tastaturfolie DM 25,-
 QUBIDE Harddisk I/F . DM 199,-

QL Anwendungen

QD Editor [V8.15] .. DM 125,-
 QMAKE [V4.13] .. DM 44,90
 QLiberator SuperBASIC Compiler DM 139,-
 QLoad-Ref DM 49,90
 QLQ [V1.13] .. DM 69,90
 QMAC Macro Assembler [V1.01] .. DM 69,-
 QMENU [V6.29] .. DM 39,90
 QPAC 1 [V1.05] .. DM 61,50
 QPAC 2 [V1.37] .. DM 119,-
 QTYP 2 Spell-Checker [V2.17] .. DM 82,50
 QPTR Pointer Toolkit [V0.28] .. DM 89,90
 QSpread Spreadsheet [V1.33] .. DM 169,-
 QSUP [V3.08] .. DM 79,90
 EPROM Manager [V3.01] .. DM 61,50
 WINED [V1.19] .. DM 49,90
 I/O 2 Toolkit [V2.15] .. DM 99,-

QL Anwendungen

LDUMP [V1.04] ... DM 65,-
NEU: DISA Intelligent Disassembler [V3.02] DM 95,-
NEU: DISA Upgrade from V1 or V2 .. DM 35,-
 EasyPTR Part 1 DM 89,-
 EasyPTR Part 2 DM 49,-
 EasyPTR Part 3 DM 49,-
 DataDesign Database DM 149,-
 DataDesign API DM 59,-
 LineDesign Vektor/DesktopPublishing DM 239,-
 Fontpack for LineDesign DM 195,-
 Pfdata DM 59,-
 Stylus-Driver für text87 und text91 .. DM 69,-
 ProWesS WindowManager + HTML Reader . DM 119,-
 HyperHelp for BASIC DM 44,90
 DiskMate 5 DM 69,-
 CueShell DM 95,-
 QDOS/SMS Reference Manual ... DM 84,90
 SerMouse Treiber DM 40,-

◀ PREISSENKUNG PREISSENKUNG PREISSENKUNG ▶

SMSQ/E Für alle Systeme V2.78

SMSQ/E ist das neue Betriebssystem mit dem Sie Ihre QL-Programme laufen lassen können und eine Unmenge neuer Möglichkeiten erhalten: schneller, flexible Diskettenformate, viele und viel schnellere BASICs und sehr viel mehr!

Für QXL & QXL 2 DM 199,-
 Für ATARIs mit QL-Emulator DM 199,-
 Für ATARIs ohne QL-Emulator DM 249,-
 Für GoldCard & SuperGoldCard DM 199,-

NEU! QXL2!

QXL2 Karte mit 8MB RAM und der "erweiterten" Version von SMSQ namens SMSQ/E für nur DM 749,- oder ohne SMSQ/E nur DM 619,-

LIEFER- und ZAHLUNGSBEDINGUNGEN

Versandkosten [Deutschland] DM 8,80 (wenn Rechnungsbetrag unter DM 50,- dann nur DM 5,80). Bei Rechnungsbeträgen über DM 500,- kostet es DM 18,- [Europa] DM 14,- (wenn Rechnungsbetrag unter DM 50,- dann nur DM 9,-). Alle Preise inkl. 15%



MwSt. Irrtum und Preisänderung vorbehalten. Verrechnungs-, Euroschecks und Kreditkarten werden akzeptiert. Bankeinzug möglich.

