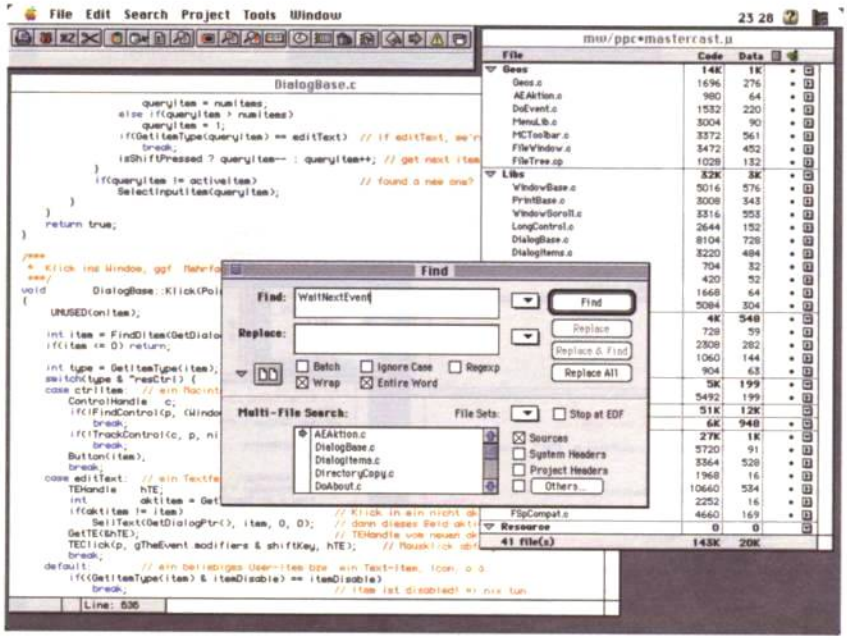

Die Projektverwaltung sowie die Editoren sind die Highlights bei den Macintosh- C-Compilern

Programmieren auf dem Mac

Compiler, Tools und Literatur

Der Einstieg in die Programmierung des Macintosh ist leider komplizierter als auf dem Atari. Dies ist im wesentlichen durch das extrem umfangreiche Betriebssystem bedingt. Ich will hier versuchen, etwas Übersicht über gängige Entwicklungs-Tools zu schaffen. Ab der nächsten Ausgabe werden wir dann auf interessante Aspekte des Betriebssystem eingehen.

Die Hardware: Die Ansprüche eines Programmierers an die Hardware sind im Vergleich zum Atari relativ hoch, im Vergleich zu Windows allerdings eher gering ...

Ein möglichst schneller Rechner mit mindestens einer 68020-CPU sollte es mindestens sein. Hier gilt natürlich die einfache Regel: je schneller, desto besser. Nötig ist ein großer Mac allerdings nicht. Aber wer wartet schon gerne auf seinen Compiler? Da man heute kaum noch Macs kaufen kann, die keinen 68040-Prozessor beinhalten, sollte dieser Punkt kein Problem sein. Ein Performa 475 oder ein Performa 630 sind prima Geräte für die Software-Entwicklung. Für die Entwicklung von nativer Software braucht man jedoch zwangsweise einen PowerMac. Für jemanden, der nur als Hobby programmiert, ist ein Compiler für nativen Code zwar nett, aber keinesfalls Pflicht [1].

Die Plattenkapazität ist für Programmierer selten ein Problem. Wer sich gut organisiert, kommt mit jeder Platte ab 80MB recht problemlos aus. Natürlich schadet ein Mehr an Platte nicht.

Nun kommen wir zum heikelsten Punkt: dem RAM-Bedarf. Während man auf einem Mac mit 4MB RAM noch ganz gut arbeiten kann, ist das Programmieren auf einem solchen Mac kaum möglich. 8MB physikalisch vorhandener Speicher sind hier das Minimum. Bei 16MB hat man schon fast zu viel RAM. Eine sinnvolle Alternative für Programmierer ist eventuell die Anschaffung des Programmes *RAMDoubler*, das aus vorhandenen 8MB quasi 16MB macht. Im Gegensatz zu virtuellem Speicher kostet es quasi keine Rechenzeit! Aber Achtung: 8MB RAM sind auch für *RAMDoubler* das sinnvolle Minimum. Eher kann man am Rechner sparen als am RAM. Beim PowerMac reichen 8MB nicht. 12MB ist dort ein sinnvolles Minimum. *RAMDoubler* verträgt sich übrigens nicht mit dem native *CodeWarrior-Debugger*.

Als sinnvolles Zubehör möchte ich noch ein CD-ROM-Laufwerk anführen. Es gibt für Entwickler sehr viel Software und Tools nur auf CD-ROMs: Da wird es rasch sehr lästig, sich ständig von Freunden das Laufwerk zu leihen. Ein ganz billiges Single-Speed-CD-ROM reicht dabei schon aus.

Die Literatur

Als nächstes ein paar Worte zur Literatur für Programmierer. Leider gibt es so etwas wie das Profibuch nicht für den Mac. Selbst deutsche Literatur für Programmierer ist quasi nicht zu finden. Zwar gibt es ein paar Einführungen in die Programmierung in Deutsch, aber die vollständige Dokumentation zum Betriebssystem existiert nur in englischer Sprache.

[1] Es gibt inzwischen Macintosh Rechner mit zwei verschiedenen CPUs: 68k-CPU, wie bei den ATARIs und dem PowerPC. 68k-Software läuft auf beiden CPUs, sie wird auf einem PowerMac per Software emuliert (die Emulation entspricht von der Geschwindigkeit in etwa einem ATARI TT). Native Software ist Software speziell für den PowerPC. Sie läuft nicht auf einem 68k-Mac.

Wer Programme für beide CPU-Typen anbieten will, erzeugt häufig ein FAT-binary-Programm. Dieses Programm enthält sowohl 68k-Programmcode, wie den entsprechenden nativen Code. Der Macintosh entscheidet automatisch, welchen Code er nehmen sollte. Die Programme sind etwa doppelt so groß, wie reine 68k-Programme, aber auf PowerMacs deutlich schneller.

Symantec ist ein alter Hase bei den Compiler-Herstellern. Den *Think C-Compiler* gibt es inzwischen in der Version 7.0.4 und eine Version 8 wird für dieses Jahr erwartet. Es gibt zwei Varianten: *Think C* und *Symantec C++*. Letztere enthält zusätzlich einen C++-Compiler. *Think C* ist ein recht flotter Compiler, der leider nur gegen Aufpreis nativen Code erzeugen kann, und dies auch nur sehr umständlich. Der Support von Symantec per EMail ist eher schwach. Preis für den Compiler: ca. 800,– DM; Schüler und Studenten zahlen deutlich unter 200,– DM. *Think C* war über Jahre – neben *MPW* – der Standard für Programmierer, bis Anfang letzten Jahres der folgende Compiler auftauchte:
Der *CodeWarrior* (kurz CW) von Metrowerks war der erste brauchbare C++-Compiler auf dem Markt, der nicht nur 68k-Code, sondern auch Native-Code erzeugen kann. Metrowerks hat – um es dezent auszudrücken – einige Anleihen bei *Think C* gemacht, was den Editor und die Projektverwaltung des Compilers betrifft. Die beiden Compiler bedienen sich nahezu identisch und sehen auch sehr ähnlich aus – nur daß der *CW* überall einen Tick durchdachter ist. Nebenbei ist der *CW* deutlich schneller als der *Think*-Compiler. Auf einem 68k-Mac ist er teilweise doppelt so schnell – auf einem Power-Mac kann er bis über 200000 Zeilen pro Minute übersetzen. Neukunden bekommen das erste Jahr die regelmäßig erscheinenden Updates frei Haus. Der Support bei Problemen ist hervorragend. Der *CW* enthält übrigens nicht nur einen C++-Compiler, sondern auch einen Pascal-Compiler. Der *CW* ist in zwei Varianten zu erhalten: Bronze und Gold. Ersterer kann keinen nativen Code erzeugen, ansonsten sind die Pakete identisch. Preis: 200,– bzw. 800,– DM. Schüler und Studenten bezahlen 100,– bzw. 200,– DM. Der *CodeWarrior* ist nur auf CD-ROM erhältlich und enthält übrigens auch die *MPW-Shell*.

Die meistbenutzten Compiler

Auf Grund der großen Verbreitung von *Think C* und dem *CodeWarrior* beschreibe ich mal etwas näher, wie diese Compiler so aussehen: neben dem üblichen Editor (beim *CW* mit farbiger Darstellung, sowie Popup-Menüs für alle Include-Files und Funktionen des aktuellen Sources) haben beide Compiler eine sehr schöne Projektverwaltung. Ein Projektfenster mit allen Sourcen ist ständig offen. Die Projektdatei entspricht den Makefiles oder der Projektdatei von *Pure C*, jedoch mit ein paar netten Features: Abhängigkeiten der Sourcen untereinander werden automatisch erkannt, man erkennt im Projektfenster immer sofort, welche Sourcen wieviel Code erzeugen. Ferner hat man - im Gegensatz zu *Pure C* – keine *.o-Files auf der Platte liegen, sondern nur seine Sourcen, die Resource-Datei und die Projektdatei – Pfade braucht man nie anzugeben, der Compiler findet die Source-Texte auch in tiefsten Ordnern. Alle Compiler unterstützen mächtige Suchoptionen, Grep ist selbstverständlich.

Der Debugger ist bei allen Entwicklungspaketen ein extra Programm, welches automatisch nachgeladen wird. Er ist mit dem *Pure Debugger* im Umfang der Features in etwa zu vergleichen. Es ist ein Sourcecode-Debugger mit der Option den Assembler-Code des Programms zu debuggen. Diverse Breakpoints lassen sich setzen und Variablen können untersucht werden.

Weitere Tools

Ebenfalls wichtig für die Programmierung ist ein Resource-Editor für Dialogboxen, Icons, usw. Dies ist eine der wenigen Punkte, wo die Macintosh Software billiger als für den ATARI ist. Das meistbenutzte Programm, der *ResEdit 2.1.3* ist nämlich kostenlos! Er ist in einigen Mailboxen zu finden, wird jedoch von den Compiler Herstellern üblicherweise mitgeliefert. Eine Anleitung in Form einer Textdatei gibt es nicht. Man muß entweder selbst die Tiefen des *ResEdit* ergründen oder aber die Anleitung beim deutschen APDA (Apple-Entwickler-Support) kaufen. Es gibt auch ein sehr brauchbares Buch zum Umgang mit dem *ResEdit*.

Ein weiteres wichtiges Tool ist ein Low-Level-Debugger. Hier bietet sich der *MacsBug* an, der – wie schon der *ResEdit* – ebenfalls von Apple stammt und ebenfalls kostenlos ist. Der *MacsBug* ist das, was der *Templemon* oder *Bugaboo* auf dem ATARI ist. Auch hier fehlt leider eine Anleitung. *MacsBug* gehört in den Systemordner und wird beim Neustart automatisch geladen. Stürzt der Mac ab, oder drückt man auf den Interrupt-Taster (Command-Power-On bei den neueren Macs), landet man im Debugger. Hier kann man mit „?" eine Befehlsübersicht bekommen.

Und was nun?

Der Einstieg in die Macintosh-Programmierung ist nicht billig, aber auch nicht besonders mühsam. Neben dem Compiler für mindestens 100,– DM fallen noch Kosten für das eine oder andere Buch und wahrscheinlich für die digitale Ausgabe der NIMs an. Dafür hat man dann aber ein komplettes und schnelles System, mit dem die Entwicklung (meistens) Spaß macht. Besonders die NIMs machen es einem leicht. Es gibt zu fast jedem Programmierproblem mit dem Betriebssystem Beispiel-Sourcen, die man lediglich übernehmen muß. Allerdings hat der eine oder andere eventuell leichte Probleme bei fast 20.000 Seiten Dokumentation den Überblick zu behalten ... trotzdem: Es lohnt sich! Welches andere Betriebssystem bietet denn z.B. fertige JPEG-Entpackroutinen an?

Wer noch mehr über Entwicklungs-Tools, Literatur oder über Macs im Allgemeinen erfahren will, dem empfehle ich das Studium der „MacFAQ", die ich einmal im Monat in der Gruppe „Mac.News" im MausNet poste. Der ca. 130KB lange Text findet sich aber auch in den Programmteilen vieler Mäuse oder in der Redaktions-Mailbox der ST-Computer. Ferner gibt es im MausNet noch die Gruppe „Mac.Dev", wo bei Programmierproblemen an sich immer jemand zur Stelle ist, der helfen kann. Sind dann immer noch Fragen offen, so kann man mich auch direkt ansprechen: *Markus Fritze @ HH2* im MausNet.

Markus Fritze

Apple-Entwicklerprogramm:
Promo GmbH
APDA-Service
Waterloohain 6-8
22769 Hamburg
Tel.: (040) 431360-66