

Im zweiten Anlauf hat Data Becker es geschafft: Nachdem das inzwischen schon etwas bejahrte Pascal 64 in verschiedenen Punkten nicht ganz zu überzeugen vermochte, liegt mit »Profi-Pascal« ein Compiler vor, den man jedem Pascal-Freund uneingeschränkt empfehlen kann. Neben dem von Kathleen Jensen und Niklaus Wirth im »Pascal User Manual and Report« beschriebenen Standard-Pascal, dessen Funktionen ohne Abstriche implementiert wurden, bietet Profi-Pascal viele sprachliche Erweiterungen, die Pascal-Freaks zum Teil aus UCSD-Pascal kennen.

Das Profi-Pascal-eigene Betriebssystem weist zwar einige kleine Schwächen auf, die jedoch ein riesiger Vorteil gegenüber anderen C 64-Programmen aufhebt: Der Zugriff auf Disketten erfolgt rund viermal schneller. Normalerweise werden 250 bis 400 Byte pro Sekunde von der Floppy in den Arbeitsspeicher (oder umgekehrt) übertragen. Unter Profi-Pascal sind es jedoch 1250 Byte pro Sekunde. Diese softwaremäßig erreichte Steigerung fällt äußerst positiv auf.

Das Hauptmenü (Bild 1) vermittelt einen Überblick über die wichtigsten Teile des Pascal-Systems. Compiler, Assembler, Editor und Betriebssystem sind auf einer Diskette gespeichert. Auf diese Diskette sollte der Programmierer achten wie auf seinen Augapfel. Sie läßt sich zwar kopieren, aber Compiler und Assembler sind dann nicht mehr lauffähig. Lediglich der Editor und die Utility-Dateien sind problemlos zu duplizieren. Pro Originaldiskette kann zusätzlich eine Sicherungskopie erworben werden. Wird die Diskette im Laufe der Zeit beschädigt oder abgenutzt, so kann sie für zwanzig Mark gegen eine Ersatzdiskette umgetauscht werden.

Obwohl Profi-Pascal mit zwei Laufwerken arbeiten kann, reicht ein Laufwerk bereits aus, um ohne häufiges Diskettenwechseln zu arbeiten, da beim Übersetzen die Quelldatei immer auf der Systemdiskette gespeichert sein muß.

## Pascal mit vielen Extras

Jedes Pascal-Programm hat denselben blockorientierten Aufbau. Der Programmblock besteht aus einem Vereinbarungs- und einem Anweisungsteil. Der Vereinbarungsteil ist wiederum in die Definition von Labels, Konstanten, Typen, Variablen und Prozeduren unterteilt. Abweichend von Standard-Pascal dür-

# Pascal für Profis

**Das neue Profi-Pascal von Data Becker enthält alle Funktionen von Standard-Pascal und darüber hinaus viele Erweiterungen. Ein weiterer Pluspunkt: Der Diskettenzugriff wurde stark beschleunigt.**

fen bei Profi-Pascal in der Konstantenvereinbarung auch einfache Ausdrücke stehen, die wiederum Konstanten enthalten. Hierzu ein Beispiel:

```
Const faktor = 3.75;
e-wert = faktor * 10 + 5;
```

Die Liste der vordefinierten Konstanten wurde um Pi (= 3.1415926536) und STKPOI als Zeiger auf den Pascal-Variablen-Stack erweitert. Der Zeiger deutet auf die unterste Speicherstelle des von oben nach unten wachsenden Variablenstack.

Die Typen Integer, Real, Boolean und Char sind standardmäßig vordefiniert. Die größte ganze Zahl ist 32767. Der größte Real-Wert lautet  $\pm 3.4028236692E+38$  und der kleinste von Null unterscheidbare Wert beträgt  $\pm 8.8162076312E-39$ . Array, Set, Record und File sind wie im Standard vorhanden. Sogar variante Records sind erlaubt. Das Attribut »PACKED« wird akzeptiert, hat jedoch keine Wirkung: Profi-Pascal speichert Daten von vornherein platzsparender, so daß Komprimieren von Werten überflüssig ist. In anderen Pascal-Versionen belegt beispielsweise eine Variable vom Typ »Char« 2 Byte, in Profi-Pascal hingegen nur 1 Byte.

Mit dem zusätzlichen String-Typ können Zeichenfolgen bis zur maximalen Länge von 132 Zeichen definiert werden. Zwischen Variablen der Typen »Array of Char« und String sind Wertzuweisungen sowie Vergleiche direkt, also ohne Indizierung möglich.

Profi-Pascal wurde um die System-Variablen »MEM« und »RANDOM« erweitert. Die Variable MEM ist so definiert, daß der gesamte Speicher des Commodore 64 als Array-Variable angesprochen werden kann. Schreiben in oder Lesen von einer Speicherstelle ist damit möglich. MEM ersetzt PEEK und POKE vollständig. Die Variable RANDOM liefert bei jeder Zuweisung eine neue Zufallszahl.

Bei der Definition von Prozeduren und Funktionen ist besonders be-

merkenswert, daß in der Parameterliste Prozeduren und Funktionen übergeben werden dürfen. Dies entspricht dem Standard, ist jedoch in kaum einer anderen Pascal-Version für Mikrocomputer vorgesehen. Man kann in Profi-Pascal also beispielsweise eine Prozedur schreiben, die Nullstellen mathematischer Funktionen berechnet. Die Funktion selbst kann dann einfach über die Parameterliste übergeben werden.

## Relative Dateien möglich

Außerdem sieht Profi-Pascal vor, Dateivariablen zu indizieren. Damit erlaubt es den wahlfreien Zugriff auf ein einzelnes, genau definiertes Element einer Datei. Damit sind relative Dateien möglich geworden, was weit über den Pascal-Standard hinausgeht, der nur sequentielle Dateien vorsieht.

Der Zugriff auf Teilbereiche eines Arrays oder Strings ohne Indizierung (!) ist vor allem bei Arrays of Char oder Stringgrößen praktisch. Beispiel:

```
Var Feld:Array(1..10) of char;
Feld(>5) = 'Sonne';
```

An die Komponenten des Arrays mit dem Namen Feld wird ab der fünften Position das Wort »Sonne« übergeben. Es paßt genau in das Array. Der Programmierer muß unbedingt darauf achten, daß der zugewiesene Wert die Grenze des Arrays nicht überschreitet.

Ein Ärgernis für viele Pascal-Programmierer ist in Profi-Pascal behoben. Die Case-Anweisung wurde um einen Else-Zweig vervollständigt. Falls für bestimmte Labels in der Case-Anweisung kein Auswahlzweig vorgesehen ist, wird die dem Else folgende Anweisung ausgeführt. Beispiel:

```
Case note of
1:Writeln('super');
2:writeln('ganz gut')
else writeln('geht schon')
end;
```

Zu den vordefinierten Prozeduren und Funktionen sind in Profi-Pascal noch eine beachtliche Anzahl hinzu gekommen. Die wichtigsten werden nun kurz erwähnt.

Über die Prozedur Allocate bestimmt der Programmierer selbst, an welchem Speicherplatz der Wert einer Zeigervariablen abgelegt wird. Bei New legt das System selbst die Adresse fest. Man muß jedoch die Aufteilung des Speicherbereichs genau kennen, da man leicht in Gefahr gerät, den Programmcode oder andere Variablen zu überschreiben.

## Programme verketteten

Continue ruft ein neues Pascal-Programm aus dem laufenden heraus auf und startet es. Ein Rücksprung in das ursprüngliche Programm wird nicht ausgeführt. Die Prozedur Execute funktioniert ähnlich wie Continue, behandelt aber das aufgerufene Programm wie ein Unterprogramm und springt nach dessen Abarbeitung in das ursprüngliche Programm zurück.

Hex gibt Integer-Größen als hexadezimale Zahlen aus, und Induc schaltet auf ein anderes Eingabegerät um, wobei die Geräteadresse anzugeben ist. Dementsprechend lenkt Outdvc die Ausgabe auf ein anderes Gerät. Setdrv bestimmt das aktuelle Arbeitslaufwerk für den Diskettenzugriff. Die Prozeduren zur Dateiverwaltung Reset, Rewrite oder Seek greifen dann auf das mit Setdrv vereinbarte Laufwerk zu. Close und Lock schließen Dateien, die im Programm mit Reset oder Rewrite eröffnet wurden. Dateien, die mit Lock gesichert werden, sind gegen Überschreiben oder Löschen geschützt. Kill löscht Dateien vom Programm aus. Seek setzt den Zugriffszeiger auf eine Datei so, daß er auf einen bestimmten Satz weist.

Dieser Satz kann dann mit Get gelesen werden.

Mark und Release steuern die Speicherplatzverwaltung auf dem Heap. Der Heap ist die Speicherplatzhalde für dynamische Variable. Nach dem Aufruf von Mark weist der Zeiger auf die oberste Position der von unten nach oben wachsenden Halde. Dann teilt die Prozedur New einer Variablen einen neuen Speicherplatz zu und die Halde ist weiter gewachsen. Release setzt den Haldenspeicherplatz wieder auf das Maß zurück, wie es mit Mark festgehalten wurde. Mark und Release ersetzen die Prozedur Dispose aus Standard-Pascal.

Die Funktion Anykey stellt fest, ob eine Taste gedrückt wurde. Getkey liefert das über die Tastatur eingegebene Zeichen. Ioerror gibt Auskunft darüber, ob bei einem Ein-/Ausgabeprozess ein Fehler aufgetreten ist.

## Overlay-Technik

Sehr lange Programme, die nicht mehr in den Arbeitsspeicher passen, können in Segmente zerlegt werden. In einem Programm dürfen maximal acht Segmente stehen. Der Compiler behandelt die Segmente so, als ob sie parallel im Speicher lie-

gen würden. Er merkt sich die Anfangsadresse des ersten Segments und compiliert die folgenden jeweils für dieselbe Adresse. Der gesamte für die Segmente reservierte Arbeitsplatz richtet sich nach dem längsten Segment. Bei der Overlay-Technik muß der Programmierer bedenken, daß sich Segmente nicht gleichzeitig im Arbeitsspeicher befinden, sondern bei Bedarf von der Diskette nachgeladen werden und jeweils die Codedatei immer zur Verfügung stehen muß. Trotzdem ist es nicht ausgeschlossen, daß sich Segmente gegenseitig aufrufen. Eine Alternative zur Overlay-Technik stellen die Prozeduren Execute und Continue sowie auch externe Unterprogramme dar.

## Externe Routinen

Externe Routinen können in Pascal oder Assembler geschrieben sein. Sie müssen vor ihrem Aufruf compiliert und an eine bestimmte Adresse im Arbeitsspeicher geladen werden. Dabei sind Speicherkollisionen zu vermeiden. An externe Unterprogramme können Werte über die Parameterliste übergeben werden.

Als weiterer Leckerbissen für den Profi können Assembler-Routinen direkt in Pascal-Programme einge-



Bild 1. Hauptmenü von Profi-Pascal und das Inhaltsverzeichnis der Systemdiskette. Systemdateien sind invers geschrieben.



Bild 2. Das »Statistical Summary« liefert Detailinformationen



Bild 3. Viele praktische Funktionen zum Bearbeiten von Dateien

fügt und gleich mitübersetzt werden. Die Anweisung Assemble in der Kopfzeile der Routine gibt dem System zu verstehen, daß eine eingebettete Maschinenroutine folgt, die mit dem integrierten Assembler zu übersetzen ist. Wie bei Pascal üblich, erzeugt der Compiler keinen reinen Maschinencode, sondern einen Zwischencode, den sogenannten P-Code, der vom Laufzeit-System während der Programmausführung interpretiert wird.

Ehe der Compiler beginnt, einen Quelltext in P-Code zu übersetzen, stellt er mehrere Fragen. Der Programmierer hat nun die Wahl, Compiler-Optionen einzustellen oder vorgegebene Werte zu übernehmen. Er kann bestimmen, an welcher Adresse die Halde und an welcher Position der Stack beginnt. Äußerst hilfreich bei der Suche nach Laufzeitfehlern ist der »Post Mortem Dump«. Sobald ein Fehler auftritt, informiert das System über die Programmstelle, an der der Fehler liegt. Man erfährt, zu welcher Prozedur oder Funktion die Zeile gehört und wie tief die Unterprogramme verschachtelt sind. Zusätzlich werden die Werte sämtlicher Variablen zum Zeitpunkt des Programmabbruchs ausgegeben. Außerdem druckt das System wahlweise den P-Code aus sowie auch ein Protokoll der Übersetzung.

Fehlermeldungen werden mit einer Nummer angezeigt. Der Programmierer muß dann im Handbuch nachlesen, um welchen Fehler es sich handelt. Nach dem Drücken der Leertaste setzt das System die Übersetzung fort. In den Editor verzweigt das System nach dem Tippen der Run/Stop-Taste. Nachdem das Programm vollständig übersetzt worden ist, stellt der Compiler noch eine kleine statistische Analyse auf (Bild 2). Sie enthält beispielsweise die Anzahl der Konstanten oder Variablen im Programm. Der Code wird automatisch unter dem Programmnamen, der in der Kopfzeile angegeben wurde, auf der Diskette gespeichert.

## Zeilenorientierter Editor

Der Editor von Profi-Pascal arbeitet zeilenorientiert. Diese einfache Lösung bietet den Vorteil, daß er wenig Platz im Arbeitsspeicher belegt und für das Programm rund 43 KByte übrig bleiben. Nachteilig ist jedoch, daß beim Einfügen von Programmteilen große Sorgfalt auf die Numerierung der Zeilen verwandt werden muß. Eine Zeile faßt 80 Zei-

chen und ist mit RETURN abzuschließen. Der Editor ähnelt ansonsten stark dem des Basic-Systems.

Der Quelltext des Editors wird mitgeliefert. Er ist bis auf einige zeitkritische Prozeduren, die in Assembler programmiert wurden, in Pascal geschrieben. So kann der Benutzer den Editor nach eigenen Vorstellungen verändern.

Profi-Pascal enthält zusätzlich noch einen Programmteil Utility (Bild 3) mit Funktionen zur Verwaltung von Dateien und Disketten. Dazu zählen das Initialisieren einer Diskette, Kopieren von Dateien und Disketten, Löschen, Laden und Umbenennen von Dateien sowie das Sichern einer Datei vor Überschreiben und die Ausgabe des Inhaltsverzeichnis.

Beim Arbeiten mit dem Utility-Paket ist Vorsicht geboten, wie auch das Handbuch anmerkt. Auf falsche Eingaben wird teilweise äußerst garstig reagiert. Erst das Drücken des Netzschalters führt zurück ins System.

Das deutsche Handbuch ist sehr ausführlich gehalten, enthält ein

Stichwortverzeichnis und ist 326 Seiten dick. Der Leser muß allerdings bereits mit Pascal vertraut sein, denn das Handbuch kann und soll kein Pascal-Lehrbuch ersetzen.

Trotz dieser kleinen Schwächen erfreut Profi-Pascal das Herz eines Pascal-Programmierers. Der Wirth-Standard wird erstmals bei einem C 64-Pascal ohne Abstriche erfüllt. Viele zusätzliche Erweiterungen ermöglichen das volle Ausnutzen aller Fähigkeiten des C 64-Systems, durch den integrierten Assembler sind sehr effiziente, maschinennahe Programme möglich. Relative Dateien verbessern die Datenverwaltung gegenüber Standard-Pascal enorm.

Alles in allem handelt es sich bei Profi-Pascal also um eine der leistungsfähigsten Pascal-Versionen überhaupt. Sowohl dem Pascal-Puristen als auch dem C 64-System-Spezialisten steht damit ein ideales Werkzeug zur Verfügung, das sich zu Recht mit dem Attribut »Profi« schmücken darf.

(Silvia Gutschmidt/ev)

Info: Profi-Pascal für Commodore 64, Data Becker, 198 Mark.

### Positiv

- Profi-Pascal entspricht Standard-Pascal. Lediglich die Pointerfunktion Dispose wurde durch Mark und Release ersetzt.
- Schneller Diskettenzugriff: 1250 Byte pro Sekunde statt der üblichen 250 bis 400 Byte pro Sekunde.
- Zeilenorientierter Editor, der wenig Speicher belegt. Es bleiben rund 43 KByte zum Programmieren.
- Quelltext des Editors wird mitgeliefert. Er kann verändert werden.
- Kompakte Speicherung auf der Diskette,
- Zugriff über »Array of Byte« auf den gesamten Arbeitsspeicher. PEEK und POKE sind dadurch überflüssig.
- Compiler liefert Post Mortem Dump, der bei der Fehlersuche sehr nützlich ist.
- Assembler-Programme können in Pascal-Programme eingefügt und gleich mitübersetzt werden.
- Random-Zugriff auf Dateien,
- Segmentprozeduren,
- Verkettung von Programmen über Continue oder Execute.
- Viele zusätzliche, nützliche, vordefinierte Typen und Funktionen.
- Der erzeugte P-Code läuft verhältnismäßig schnell.
- Ausführliches Handbuch
- Backup- und Ersatzdisketten erhältlich

### Negativ

- Systemdiskette nicht kopierbar.
- Beim Compilieren muß die Quelldatei auf der Systemdiskette gespeichert sein. Bei häufiger Benutzung führt dies zu einer raschen Abnutzung der Systemdiskette.
- Im Inhaltsverzeichnis sind Text- und Codedateien nicht gekennzeichnet.
- Das System bringt keine Warnung, wenn eine bereits existierende Datei überschrieben wird.
- Meldungen über Laufzeit- und Syntaxfehler sind teilweise schwer verständlich.
- Systemmeldungen in englisch.