Aktion mit Action

Action zählt zu den schnellsten Programmiersprachen für Atari-Computer. Wer auf hohe Geschwindigkeiten Wert legt, aber nicht in Maschinensprache programmieren will, sollte Action in Erwägung ziehen.

In letzter Zeit machte eine Programmiersprache von sich reden, die ursprünglich als Sprache für Systementwickler konzipiert war: »C«. Sie wurde Anfang der siebziger Jahre von Dennis Ritchie aus BCPL weiterentwickelt. Als Standardwerk zur Programmierung in »C« wird daher auch oft das Buch von Kernighan/Ritchie (»The C Programming Language«) genannt. Ritchie war später auch an der Entstehung von UNIX beteiligt.

Es gab bereits mehrere Versuche, die Sprache C auf den Atari anzupassen: Das »Deep Blue C«-System ist weitgehend an einen klassischen C-Compiler angelehnt. Allerdings sind hier nicht nur beim Befehlsumfang, sondern auch bei der Arbeitsgeschwindigkeit Abstriche zu machen. Ein anderer Weg wurde mit C/65 begangen. Bei diesem System wird der Quelltext in einen Assemblertext compiliert, der dann mit MAC/65 (siehe Beitrag in diesem Sonderheft) weiter bearbeitet werden muß.

Unser Thema ist aber die Programmiersprache »Action«, die von OSS stammt. Diese Sprache enthält zwar viele Elemente von C, ist aber dennoch »anders« genug, um einen eigenen Namen verdient zu haben.

Action wird (womit wir schon beim ersten Unterschied zu konventionellen C-Compilern sind) auf einem Programmmodul geliefert. Wie bei allen anderen Cartridges von OSS handelt es sich dabei um ein 16 KByte-Super-Cartridge. Es belegt aber trotzdem nur 8 KByte-RAM-Speicher. Bei Verwendung von DOS XL kann der freie Speicherplatz sogar noch weiter erhöht werden. Der nächste Vorteil ist (genau wie bei MAC/65), daß sich Editor und Compiler gleichzeitig im Speicher befinden. Von der Konzeption des Systems her läßt sich Action also mit Turbo-Pascal vergleichen, bei dem ebenfalls Editor, Compiler und Quelltext gleichzeitig im Speicher vorliegen. Ein separater Linker ist nicht nötig.

Da Action, genau wie C, nicht mit Zeilennummern arbeitet, hat man dem Action-System einen stark an Textprogramme angelehnten Editor eingebaut.

Grundsätzlich stehen alle normalen Fähigkeiten des Bildschirmeditors zur Verfügung. Allerdings kann jede Zeile maximal 240 Zeichen lang sein, so daß man bei Schleifen und Kommentaren genügend Platz zum Einrücken hat. Da die Darstellung auf dem Bildschirm auf nur 40 Zeichen pro Zeile begrenzt ist (Bild 1), ist natürlich immer nur ein Teil des Listings zu sehen. Bei den Zeilen, die über den Rand des Bildschirms hinausreichen, ist jeweils das letzte Zeichen invertiert. Bewegt man nun den Cursor über den rechten Rand des Bildschirms, wird die Zeile, in der man sich befindet, nach links gescrollt (Bild 2). Weitere Fähigkeiten des Editors sind Funktionen zum Suchen und Ersetzen von Begriffen, der Einsatz eines zweiten Bildschirmfensters, in dem man eine andere Datei bearbeiten kann, und ähnliches mehr.

Das Fehlen von Zeilennummern beim Programmieren erschwert es oft, bestimmte Programmteile wiederzufinden. Der Action-Editor erlaubt aber, an beliebigen Stellen Markierungen (Tags) zu setzen, die man dann später mit nur zwei Tasten wieder anwählen kann.

Sämtliche Parameter wie Zeilenlänge und Fenstergröße etc. sind veränderbar.

Den Mittelpunkt des Action-Systems bildet der Monitor, von dem aus man Editor, Compiler und DOS aufrufen kann. Außerdem können einzelne Speicherzellen verändert und ganze Speicherbereiche gelistet werden. Wichtig beim Testen eines Programms ist der TRACE-Modus. Während des Programmablaufs erlaubt er die Ausgabe von Namen und Parametern jeder aufgerufenen Prozedur auf dem Bildschirm.

Die Programmiersprache Action ist weitgehend an C angelehnt. So haben beide grundsätzlich ähnliche Datentypen und eine ähnliche Syntax.

Da zwischen den einzelnen Befehlen nur Leerzeichen stehen brauchen, ist man in der Aufteilung des Quelltextes völlig frei.

Natürlich ist es empfehlenswert, bei Verschachtelungen, wie bei anderen Programmiersprachen auch, dazwischenliegende Zeilen einzurücken. So werden Programme übersichtlich und überschaubar.

An Datentypen sind in Action zunächst einmal BYTE und CHAR (einzelne Byte), INT (2-Byte-Integer zwischen -32768 und 32767) und CARD (desgleichen zwischen 0 und 65535) vertreten. Bei der Deklaration von Variablen kann man (muß aber nicht) angeben, welche Speicherstelle dazu benutzt werden soll. Beispiel:
»BYTE chbase=$02F4«

Mit dieser Zeile würde man eine 1-Byte-Variable erzeugen, die genau auf der Adresse $02F4 liegt. Bei einer Wertzuweisung zu chbase wird die Adresse $02F4 geändert. Das Kommando »chbase=$CC« schaltet also den internationalen Zeichensatz ein. $02F4 fungiert nämlich als Basisregister für den Zeichensatz und $CC als High-Byte der Anfangsadresse des Zeichensatzes. Wo Basic umständliche POKE-Befehle erfordert, kann man hier sehr viel eleganter programmieren. Gleiches gilt natürlich für die anderen Datentypen.

Neben den fundamentalen Datentypen gibt es in Action auch erweiterte Arten von Variablentypen. Fangen wir mit dem Typ POINTER (Zeiger) an. Davon abgesehen, daß die Deklaration ein wenig anders abläuft (»CHAR POINTER pnt« statt »CHAR *pnt«), ist die Anwendung der Pointer in Action genauso wie in C gelöst. Mit »pnt=@var« erhält man die Adresse einer Variablen (in C:»pnt=&var«), mit »pnt ^« erhält man die Adresse, auf die pnt zeigt (in C müßte der Befehl dann »*pnt« lauten).

Ein weiterer wichtiger Datentyp ist das Feld (Array). Im Gegensatz zu C verwendet man in Action allerdings nur eindimensionale Arrays. Sie werden außerdem nur aus den fundamentalen Datentypen, also CHAR, BYTE, INT und CARD, aufgebaut. Allerdings kann man mit einem Trick auch Arrays aus den erweiterten Datentypen verwenden.

In Action lassen sich auch eigene Datentypen definieren; sie heißen Records. Man benutzt dazu die Anweisung TYPE, während es in C STRUCT heißt. Der Vorteil der Records liegt darin, verschiedene Datentypen unter einem gemeinsamen Oberbegriff zusammenfassen zu können.

Wie schon bei der Erörterung der fundamentalen Datentypen zu sehen war, erlaubt Action eine sehr maschinennahe Programmierung. Alle numerischen Konstanten können auch in hexadezimaler Schreibweise eingegeben werden, und auch sämtliche Funktionen zur Manipulation einzelner Bits sind verfügbar: »&« (logisches AND), »%« (OR), "!" (XOR), LSH (left shift) und RSH (right shift). Leider fehlt eine angenehme Eigenschaft von C. Es ist nicht möglich, Variablen an beliebigen Stellen im Programmtext durch zwei kurze Zeichen zu in- oder dekrementieren.

Immerhin kann man bei Veränderungen einer Variablen den Namen auf der rechten Seite des Gleichheitszeichens weglassen, vorausgesetzt man schreibt stattdessen zwei Gleichheitszeichen (Beispiel: »Zaehler==+1« erhöht die Variable Zaehler um eins).

Ein wichtiger Gesichtspunkt bei der Beurteilung einer Sprache sind die verfügbaren Strukturen für Schleifen und Verzweigungen. Action ähnelt in dieser Hinsicht weniger C, sondern eher Basic oder Pascal. Für Verzweigungen dient natürlich das Kommando IF. Hierbei ist die Anzahl der Befehle, die THEN folgen dürfen, völlig offen. Am Ende des durch THEN eingeleiteten Blocks muß allerdings ein FI stehen. ELSE ermöglicht es dann, den entgegengesetzten Fall abzufangen. Die Worte DO und OD beginnen beziehungsweise beenden einen Schleifenblock. Diese beiden Ausdrücke ersetzen die geschweiften Klammern, die üblicherweise zwischen einfachen Anführungszeichen eingebunden sind. Dabei kann man ohne Bedingungen arbeiten und erhält dann eine unendliche Schleife - oder aber WHILE, FOR oder UNTIL benutzen, um eine Bedingung für die Schleife festzulegen. Einen vorzeitigen Abbruch von Schleifen bewirkt der Befehl EXIT.

Was eine strukturierte Programmiersprache sein will, muß natürlich über Prozeduren, Funktionen und lokale Variablen verfügen. Während sich Prozeduren und Funktionen in C nur darin unterscheiden, daß Funktionswerte zurückgegeben werden (oder nicht), werden sie in Action bereits bei der Deklaration unterschieden.

Bei Prozeduren geht das folgendermaßen vor sich: Zunächst wird mit PROC angezeigt, daß eine Prozedurdeklaration stattfindet. Es folgen der Name der Prozedur und (in Klammern) Datentypen und Namen der übergebenen Parameter. Damit sind die übergebenen Parameter automatisch als lokale Variablen für diese Prozedur definiert.

Beendet wird der Vorgang mit dem Kommando RETURN. Prozeduren werden wie Kommandos aufgerufen (Beispiel: »Wait (30)«).

Funktionen unterscheiden sich von Prozeduren darin, daß sie einen Funktionswert an das aufrufende Programm zurückliefern können. Ein Aufruf einer Funktion sieht deshalb auch ein wenig anders aus. Beispiel: »X=STICK (0)«. Bei Funktionsdeklarationen muß zunächst der Typ des ermittelten Endresultats angegeben werden. Es folgt dann das Kommando FUNC, der Name der Funktion und die Liste der übertragenen Parameter (genau wie bei Prozeduren). Der einzige weitere Unterschied ist, daß zum Schluß dem Befehl RETURN noch ein Parameter übergeben wird, der dann als Funktionswert gilt.

Genau wie bei einem C-System werden auch zum Action-Compiler die wichtigsten Ein- und Ausgabefunktionen in Form einer Unterprogrammbibliothek mitgeliefert. Der große Unterschied ist allerdings, daß diese Routinen nicht, wie sonst üblich, auf Diskette beiliegen und dann in das compilierte Programm eingebunden werden, sondern im ROM des Action-Cartridges liegen. Dies hat zur Folge, daß compilierte Programme, wenn sie auf vordefinierte Funktionen oder Prozeduren zurückgreifen, nur mit eingestecktem Cartridge laufen.

Weiterhin gibt es, passend zum Action-Modul, eine Unterprogrammbibliothek. Darin enthalten sind zunächst Befehle zur Ausgabe sämtlicher Datentypen (inklusive vor PrintF). Daneben findet man, auch jeweils für alle Datentypen, den INPUT-Befehl. Ebenso sind alle anderen normalen Ein- und Ausgabefunktionen, wie GET, PUT, OPEN, CLOSE, XIO, NOTE und POINT, bereits als Prozedur definiert. Leider sucht man die recht wichtigen Befehle BGET und BPUT vergeblich. Grafik- und Soundbefehle, die zum größten Teil ihren Basic-Vorbildern entsprechen, fehlen aber nicht.

Da Action nicht ohne weiteres in der Lage ist, Zeichenketten zu verarbeiten, sind auch hierfür einige nützliche Prozeduren vordefiniert.

Wie wir bisher gesehen haben, hat Action zwar viele Eigenschaften von C, eine vollständige Implementation liegt aber beileibe nicht vor. Andererseits wartet Action mit Erweiterungen auf, die besonders auf dem Atari von sehr großem Nutzen sind. Dazu zählt zum Beispiel auch die Festlegung der Adresse einer Variablen.

Wer zeitkritische Programmteile doch lieber in Maschinensprache schreiben will, dem stehen zweierlei Wege offen. Für kurze, verschiebbare Programmteile empfiehlt es sich, den Maschinencode direkt als Datenblock in den Quelltext einzubinden. Längere Programme kann man ab einer festen Adresse assemblieren und die zugehörige Prozedur im Quelltext einzig und allein durch Festlegung ihrer Anfangsadresse definieren.

Für den professionellen Programmierer ist es wichtig zu wissen, daß er die volle Kontrolle darüber hat, wie der Compiler den Speicherplatz aufteilt. So läßt sich dann der erzeugte Code in ein EPROM brennen.

Weiterhin ist für den ernsthaften Einsatz wichtig, die Verwendung der im ROM integrierten Unterroutinen unterbinden zu können. Man definiert einfach die Routinen selbst neu. Einfacher ist es allerdings, auf die Action-Runtime-Library zurückzugreifen. Der Anwender bekommt eine Reihe vordefinierter Dateien, die mit dem INCLUDE-Befehl in die selbstverfaßten Programme eingebunden werden.

```
;      Erase8(127-x1+y0, 127+y1)
    FI
  FI
RETURN

PROC GetParam(STRING param, CARD ARRAY c
  CARD resultC
  STRING numBuf(0)=$550

  Print(param)
  Print(" = ")
  PrintC(cur^)
  IF initial THEN
    Print(", initial = ")
    PrintCE(initial^)
  ELSE
    PutE()
  FI

  Print("Enter new value:  ")
  resultC = InputC()
  IF numBuf(0)#0 THEN
ACTION! (C)1983 ACS
```

Bild 1. Eine Zeile kann bis zu 240 Zeichen lang sein. Da sich nur 40 Zeichen auf dem Bildschirm darstellen lassen, wird die Zeile, in der sich der Cursor befindet, gegebenenfalls nach links oder rechts gescrollt.

```
;      Erase8(127-x1+y0, 127+y1)
    FI
  FI
RETURN

(STRING param, CARD ARRAY cur, initial)
  CARD resultC
  STRING numBuf(0)=$550

  Print(param)
  Print(" = ")
  PrintC(cur^)
  IF initial THEN
    Print(", initial = ")
    PrintCE(initial^)
  ELSE
    PutE()
  FI

  Print("Enter new value:  ")
  resultC = InputC()
  IF numBuf(0)#0 THEN
ACTION! (C)1983 ACS
```

Bild 2. Die mit dem Cursor gekennzeichnete Zeile ist bereits nach links gescrollt. Alle anderen Zeilen befinden sich an der ursprünglichen Position. So bleibt der zu bearbeitende Text übersichtlich.

Unter dem Strich gesehen handelt es sich bei Action (etwa 250 Mark) um eine professionelle Programmiersprache mit Arbeitsgeschwindigkeiten, die auf dem Atari bisher für eine höhere Programmiersprache als unerreichbar galten: Ein Benchmarkprogramm zur Berechnung der ersten 1000 Primzahlen war in Action nur 40 Prozent langsamer als die in Maschinensprache geschriebene Version.

Ein Grund für diese enorme Geschwindigkeit ist darin zu suchen, daß der Compiler intern völlig anders vorgeht als ein normaler C-Compiler. Durch diese vereinfachte maschinensprachnähere Arbeitsweise ist leider auch die Fähigkeit verlorengegangen, rekursiv zu arbeiten.

Als weiteren Zusatz zu Action bekommt man für etwa 100 Mark die »Programmer's Aid«-Diskette. Auf dieser Diskette findet man dann schmerzlich vermißte Befehle wie CIRCLE, Kommandos für Player/Missile-Grafik oder Fließkommaverarbeitung sowie einige Demoprogramme.

Action hat also für jeden etwas zu bieten: Dem Umsteiger von Basic wird eine schnelle, komfortable Sprache geboten, die auch dem Profi-Programmierer noch voll ausreicht. Und wer später plant, auf einem »großen« Computer in C weiterzumachen, kann sich bereits mit seinem Atari einarbeiten.

(Julian F. Reschke/wb)

Bezugsquellen:
CompyShop, Gneisenaustr. 29, 4330 Mühlheim/Ruhr,
Tel. (0208) 497169
Münzenloher, Tölzer Str. 4, 8150 Holzkirchen,
Tel. (08024) 1814

Das Textverarbeitungs-Sextett

Sechs verschiedene Textverarbeitungen auf dem Prüfstand. Welche ist die beste?

Den Heimcomputer als Schreibmaschine zu nutzen ist wohl eine der häufigsten ernsthaften Anwendungen im Heimbereich. Insgesamt sechs verschiedene Textverarbeitungen sollen deshalb verglichen werden, damit Sie beim Kauf die richtige Wahl treffen.

Eine der ältesten Textverarbeitungen für den Atari ist wohl der »Atari-Schreiber«. Er ist auf Modul und neuerdings auch auf Diskette lieferbar. Der »Atari-Schreiber« ist die deutsche Version des amerikanischen »Atari-Writers« und deshalb auch auf deutsche Verhältnisse abgestimmt. Sowohl das Handbuch wie auch die Benutzerführung im Programm sind in deutscher Sprache abgefaßt. Ebenso ist der Zeichensatz und sogar die Tastaturbelegung der deutschen Norm angepaßt. Wer sonst viel mit der Schreibmaschine umgeht, wird darüber erfreut sein. Alte Hacker auf dem Atari können sich allerdings anfangs wahrscheinlich vor Tippfehlern nicht mehr retten. Deshalb liegen dem Programm auch Aufkleber für die Tastatur bei, die die umbelegten Tasten entsprechend kennzeichnen.

Die Bedienungsanleitung zum »Atari-Schreiber« besteht aus drei Teilen. Das eigentliche Handbuch beinhaltet eine ausführliche Beschreibung des Programms und gibt Tips zum Umgang mit dem System. Zudem liegt eine Referenzkarte bei, in der alle Befehle von »Atari-Schreiber« kurz, aber informativ aufgeführt sind. Zum schnellen Nachschlagen von bestimmten Befehlen benutzt man am besten die Referenzkarte, die sich als nützliches Hilfsmittel beim Umgang mit der Textverarbeitung erweist.

Der »Atari-Schreiber« wird von einem Hauptmenü aus gesteuert, das die wesentlichen Funktionen zur Bearbeitung eines Textes enthält (Bild 1). Dieses Menü erscheint sofort nach dem Laden des Programms und erleichtert so den ersten Einstieg in die Textverarbeitung. Auch ohne lange im Handbuch nachschlagen zu müssen, kann man gleich mit der Eingabe des ersten Textes beginnen. Man betätigt einfach die Taste »N« und gelangt in den Schreib-Modus. Am oberen Bildschirmrand ist nun eine Zeile zu sehen, welche die grundlegenden Druckparameter, wie linke und rechte Randbegrenzung, oder den Zeilenabstand enthält. Am unteren Bildrand befindet sich die Anzeige über den aktuellen Cursorstand. Leider beziehen sich diese Angaben nur auf die Bildschirmposition und nicht auf die tatsächliche Position des laufenden Textes. So ist eigentlich nur die Spaltenangabe von Nutzen. Insgesamt lassen sich 21 Zeilen von je 35 Zeichen auf einmal auf dem Bildschirm darstellen. Ein horizontales Scrollen über die 35. Spalte hinaus findet nicht statt. Die Darstellung von Tabellen kann daher zum Problem werden. Die Formatierung eines Textes auf dem Bildschirm ist nicht identisch mit dem späteren Ausdruck. Der Text wird nur dem Bildschirmformat angepaßt; von den Einstellungen, die Ränder, Blocksatz und ähnliches betreffen, muß man sich leider erst auf dem Papier überraschen lassen.

Die Druckeranpassung ist für die Atari-Drucker 1020, 1025 und 1027 sowie für den Epson FX-80 über ein kleines Untermenü vorzunehmen. Andere Drucker werden vom Atari-Schreiber nicht unterstützt, so daß man in diesem Fall entweder für entsprechende Steuerzeichen im Text sorgen muß oder sich selbst ein passendes Drucksteuerprogramm schreibt.

Beim »Atari-Schreiber« ist die Ausgabe von Serienbriefen vorgesehen. Daher kann man einen Text auch mit Variablen versehen. Braucht man beispielsweise ein Rundschreiben mit gleichlautendem Text, aber persönlicher Anrede, so wird die Anrede durch eine Variable ersetzt. Beim Ausdruck des Rundschreibens schließlich steht anstatt der Variablen der eigentliche Text. Dieser wird einfach einer Datei von der Diskette entnommen, die die erforderlichen Daten enthält.

Stern oder Sternschnuppe?

Wer größeren Wert auf komfortable Benutzerführung legt, ist mit dem »StarTexter« gut beraten. Allerdings meldet sich »StarTexter« mit lautem Getöse und grafischen Spielereien. So würde man eher ein neues Spiel an Stelle eines Textprogramms vermuten. Doch der erste Schein trügt. Wie schon der »Atari-Schreiber« ist auch dieses Programm auf den deutschen Benutzer zugeschnitten. Man verfügt also über einen deutschen Zeichensatz, kann aber auch beliebige andere Zeichensätze nachladen. Auf der Diskette ist deshalb ein spezielles Programm mit dem Namen »StarFont« enthalten. Damit kann man sich seine eigenen Zeichen definieren. Auf diese Art lassen sich natürlich auch spezielle grafische Symbole in einen Text aufnehmen, denn alle Sonderzeichen werden nicht nur auf dem Bildschirm dargestellt, sondern auch in dieser Form auf dem Drucker ausgegeben. Die Tastaturbelegung kann man sich aussuchen. Entweder arbeitet man mit der regulären Belegung der Tasten, oder man wandelt die Tastatur des Atari in eine deutsche Schreibmaschinentastatur nach DIN um, je nach eigenen Bedürfnissen und Gewohnheiten.