

oder Kombinationstasten werden nicht zu umgehen sein.

Viele Textverarbeitungsprogramme benutzen entweder intern schon einen 8-Bit-Code oder sind wenigstens auf die Möglichkeit vorbereitet (z. B. Wordstar 3.4). Andere sind auf den 7-Bit-Code beschränkt und aus mehreren Gründen nicht ausbaubar, zum Beispiel, weil die Entwickler das achte Bit als Kontroll-Flag für die Formatierung benutzen (z. B. Wordstar 3.1). Ein Entscheidungskriterium für die Beschaffung eines Textverarbeitungssystems ist deshalb ab sofort auch, ob es den erweiterten Code verarbeitet.

Leider sind die Besonderheiten in der Schreibweise der osteuropäischen Sprachen mit lateinischer Schrift in der neuen Norm noch nicht berücksichtigt. Das ist ein Handicap für Firmen und Institutionen, die wirklich gesamteuropäisch arbeiten. Die 96 zusätzlichen druckbaren Zeichen des 8-Bit-Codes reichten nur für die mittel-, west- und nordeuropäischen Sprachen aus, wobei die geringe Verdichtungsrate von Computern im Ostblock sicher mitentscheidend für

Lage der wichtigsten Sonderzeichen im 8-Bit-Code

dez.	hex.	Zeichen
167	A7	\$
169	A9	©
174	AE	@
177	B1	±
178	B2	?
179	B3	'
181	B5	µ
196	C4	À
214	D6	Ö
220	DC	Û
223	DF	ß
228	E4	ä
246	F6	ö
252	FC	ü

die Auswahl war. Die Bezeichnung der ISO-Norm weist darauf hin, daß weitere Alphabete geplant sind, die wahrscheinlich auch die osteuropäischen Sprachen mit lateinischer Schrift abdecken. Dann ergeben sich wieder die gleichen Schwierigkeiten wie mit den nationalen Versionen des 7-Bit-Codes: siehe oben.

nen Kommandobuchstaben einlesen will, denn ohne Return geht das eben nicht.

In diesem Fall hilft das Programm in Bild 1 weiter: Hier wird mit dem read-Befehl immer nur ein eingegebenes Zeichen eingelesen. Einen Pferdefuß hat allerdings auch diese Lösung: Mit einem CTRL-Z (= char(26)) wird in CP/M immer das Ende einer Textdatei angezeigt, und gibt man dieses Zeichen von der Tastatur ein, so tritt eine Fehlermeldung auf (ret = -1), die Datei wird geschlossen und man kann von der Tastatur überhaupt keine Zeichen mehr eingeben.

Verwendet man in einem C-Programm die Standard-Bibliothek (stdio.h) und bindet mit LO68 sowohl CLIB als auch LIBF.A mit diesem Programm zusammen, so erscheint die (ansonsten folgenlose) Fehlermeldung

Undefined Label(s):
__optoff

Die Datei OPTOFF.O ist jedoch in der CLIB-Bibliothek enthalten. Wenn die Fehlermeldung also stört, der kann sie folgendermaßen beseitigen: Zunächst holt man mit dem Archiv-Programm AR68 die Datei OPTOFF.O aus CLIB und legt sie unter ihrem eigenen Namen als neue Datei auf der Disk ab. Sodann wird, wieder mit Hilfe von AR68, diese Datei auch in LIBF.A eingebunden. In Bild 2 sind die notwendigen Befehle aufgeführt.

Kr.

C-Programme unter CP/M-68k

Die Dokumentation des C-Compilers unter CP/M-68k im Handbuch von Digital Research ist zwar teilweise recht gut, manche Aspekte werden aber nur sehr am Rande erwähnt. Das betrifft zum Beispiel die Beschreibung des getchar-Befehls aus der Standard-Bibliothek. Laut Beschreibung holt diese Routine ein eingegebenes Zeichen vom aktuellen Standard-Eingabekanal, im Normalfall also von der Tastatur.

Tatsächlich jedoch legt die Routine beim ersten Aufruf alle im Folgenden von der Tastatur eingegebenen Zeichen in einem Puffer ab, und zwar so lange, bis ein Return eingegeben wird. Erst dann wird das erste eingegebene Zeichen an das aufrufende C-Programm weitergeleitet. Beim nächsten getchar-Aufruf erhält man das zweite Zeichen und so fort; statt des ursprünglich eingetippten Return (= char(13)) wird aber mit einem Line-Feed (= char(10)) das Ende der Zeile angezeigt, weil dies der in Unix-Systemen üblichen Darstellung entspricht.

So angenehm die Bereitstellung eines Puffers in einigen Fällen auch ist – so wird zum Beispiel bei Delete das falsch eingegebene Zeichen direkt in diesem Puffer gelöscht – so unangenehm er-

weist sich dieses Verhalten, wenn man etwa in einem Menü nur einen einzel-

```

main()
{
    char kommando;
    ...
    printf("Weiter ? j/n !");
    while((kommando = getkom()) == 'j')
    {
        ...
    }
    ...
}

getkom()
{
    char ch;
    int ret;

    ret = read(0,&ch,1);
    if (ret != 1)
    {
        printf("Unzulaessige Eingabe, Programm-Abbruch");
        exit();
    }
    return(ch);
}

```

Bild 1. Eine Ja-/Nein-Abfrage läßt sich so auch ohne folgendes Return beantworten

```
A>AR68.REL W CLIB OPTOFF.O >OPTOFF.O
```

```
A>AR68.REL R LIBF.A OPTOFF.O
```

Bild 2. Nach Durchführung dieser Änderung ist __optoff auch in LIBF.A definiert