

EDV FÜR DEN ARCHITEKTEN (ODER WIE SIEHT DER NÄCHSTE WINTER AUS?)

Nun also doch - man spricht darüber. Und viele Fragen, Kaufen oder Leasen? Welches System? CAD, da sind wir doch eigentlich zuständig? Ist programmieren schwer?

Erfindungen werden gemacht, wenn sie gebraucht werden. Die Erfindung der EDV liegt 40 Jahre zurück, die Entwicklungen zu den heutigen Systemen hin haben in den letzten 30 Jahren stattgefunden, weitgehend in den USA. Eine gern als Marktführer bezeichnete Firma hat 1983 in Europa fast 11 Mrd Dollar umgesetzt, Siemens als drittgrößter 1,4. Stürmisch entwickelt sich das jüngste Kind, die Microbranche: in der BRD standen am Jahresanfang 1 Mio 'Anlagen', in diesem Jahr werden 650 Tausend dazukommen, 1986 werden es 2 Mio. sein. Das Wachstum auf diesem Gebiet soll aber, so Diebold, noch mindestens 4 Jahre lang bei über 30% bleiben, denn Gewerbe und Mittelstand haben sich bisher zurückgehalten.

Die kleineren Unternehmen sind demnach nicht die gewesen, von denen die bisherige Entwicklung getragen wurde; sie haben also, grob vereinfacht, die enormen Kosten gespart, die für diesen Fortschritt aufzubringen waren, für Lochkartensysteme mit schwerer Mechanik, für einfache Programme in zu schwierigen Sprachen, für aufwendige Schulung, Bedienung und Wartung. Können sie nun froh sein, haben sie überholt ohne einzuholen?

Die Rechnung der Großen sieht ganz anders aus, und geht bald auf. Sie haben ihre umfangreiche Verwaltung maschinisiert, die Leistungskontrolle des Einzelnen neu aufgebaut, zunehmend Arbeitsplätze für Hochschulabsolventen geschaffen, haben die eigene Entwicklungstätigkeit verstärkt, und den wissenschaftlichen Fortschritt schneller erkannt und umgesetzt. Die Kosten aber waren in den langen Aufbaujahren wohl zu tragen, durch andere.

Nach dieser Veränderung sind sie nun Wirtschaftsträger geworden, die sich nicht einfach nur durch ihre Größe von den anderen, bisherigen unterscheiden. Es ist auch nicht allein der tiefgreifende Umbau der inneren Verwaltung, mit der sie nun ohne vergrößerten Überbau in höhere Umsatzklassen vorstoßen. Es ist vielmehr ihre neue Fähigkeit, bisher nur unzusammenhängende Marktaussagen zu erfassen und auszuwerten sowie die andere, sich planerisch und ggf. fertigungstechnisch schnell einzustellen und Nischen tiefer auszufischen. Damit wachsen den größeren Unternehmen 'gestalterische' Kräfte am Markt zu. Deutlicher als je zuvor ist neben die Größteilung zusätzlich die nach Arten getreten.

Ist hier eine Entscheidung gewollt, eine Wahl überhaupt erkannt worden? Mindestens 'unten' bei den kleinen Mitgliedern wohl nicht. Ihnen hat ja nicht nur das Kapital gefehlt für Anschaffung und Unterhalt, sondern auch die Einsicht, und der Bedarf. Und dies alles hat sich nun geändert.

Was aber soll sich jetzt bei Architekten geändert haben? Eigentlich auch alles. Die EDV-Preise sind vorstellbar geworden, ohne in einem ausstattungslosen Beruf die Lustschwelle zu unterschreiten; diese Lust ist die ledige Mutter der Einsicht. Und auf der Bedarfsstelle ist bei über 40 m² pro Kopf auch wesentliches schon geschehen, und es muß auf neue, d.h. alte Ideen dort gesetzt werden - bleibt also die Frage, ob die auch EDV-gerecht sind.

Was tun? Möglichst nichts: denn nirgendwo so deutlich wie hier, auf dem Felde der Datenverarbeitung zeigen sich die Folgen von Fähigkeiten und Konzepten. Dieser Beitrag soll deshalb nicht den modernen Verbraucher an die 'neuen Möglichkeiten' heranzuführen, sondern ihm Mut machen, irgendwo nach Lust anzufangen und daraus einen Weg zu Fähigkeiten werden zu lassen. Erst die Fragen, die sich dann stellen, führen weiter.

Das 'System'

besteht aus einem Kasten mit Elektroanschluß, einem Nahseher mit Schreibtastatur, einem Drucker, und möglichst vielen und guten Programmen.

Die Programme sind was der Name sagt, geordnete Abläufe: Daten, die irgendwann über die Tastatur eingegeben worden sind, sollen in gewollter Reihenfolge ausgesucht, verändert (oder nicht), und ausgegeben werden.

Der E-Kasten besteht aus einem Rechenwerk (dem zentralen Prozessor, CPU), einem Arbeitsspeicher, und meist zugehörig einem oder mehreren externen Speichern. Speicher sind wie ein Rechenblatt, Räume mit nummerierten Zellen. In jede Zelle paßt ein Buchstabe oder eine Zahl. Für große oder Zahlen mit Komma nimmt man mehrere benachbarte Zellen. Der Arbeitsspeicher ist der, aus dem heraus und in den hinein die CPU arbeitet, die den eigentlichen Computer darstellt. Wird der E-Kasten, die Zentraleinheit (ZE) abgestellt, sind alle Programme und Daten im Arbeitsspeicher verloren; um dies bei Stromausfall oder Blitzschlag zu verhindern sind alle größeren Anlagen über einen Akku an das Netz angeschlossen.

Externe Speicher sind eigene mechanische Baugruppen ('Laufwerke'). Sie schreiben oder lesen auf magnetisierbare Träger (Band oder Platte) - die Daten werden durch die Magnetisierung zu einem dauerhaften und doch änderbaren Muster. Nach Abschalten der Anlage bleiben die Daten, das ist der Zweck der externen Speicher, sehr wohl erhalten, sie können sogar mit dem Datenträger aus der Maschine genommen und auswärts gelagert werden (Sicherung). Freilich kommt der Rechner (die CPU) an die Daten in dem externen Speicher nur auf dem Umweg über den Arbeitsspeicher heran, in dem sie erst gelesen werden müssen. Über den Grund dafür etwas später.

Mit 'Daten' sind hier nicht nur die Eingaben des Benutzers gemeint, sondern auch die Programme, die diese Eingaben verarbeiten sollen. Beide sind als 'Dateien' auf dem Datenträger abgelegt, und können nur unter ihrem Namen aufgerufen werden. Dabei unterscheiden sich Band und Platte wesentlich. So wie man im Heimkino auf die bestimmte schöne Stelle nicht direkt zugreifen kann, sondern an den Anfang spulen muß, dann im Schnellauf einzelne Stellen sucht bis zum Finden, so kann auch das Band nur sequentiell gelesen werden. Schwieriger noch wird das beim Rückschreiben, weil die rückzuschreibende Datei vielleicht länger geworden ist, und nun die folgende Datei überschreiben müßte - im Microbereich gibt es meist nur Platten, und auf die wird unzusammenhängend (in 128 oder 256 Stellen langen 'Sektoren') geschrieben, wo gerade ein freier Sektor sich findet; der Benutzer merkt dabei von der 'Plattenverwaltung' nichts. Erwähnt sei noch eine Verschiedenheit der Plattenausführung in diesem Kleinbereich, die leicht wechselbare 'Floppy Disk', und die größere, schnellere, sichere, aber bisher meist nicht wechselbare 'Festplatte'.

Will man also mit der Anlage arbeiten, so schaltet man sie ein, liest das Programm von der Platte, das dann die Benutzerdaten vom Bildschirm (BS) abfordert oder, wenn schon vorher eingegeben, von der Platte liest. Die Abforderung von Fall zu Fall nennt sich 'interaktiv', das andere 'Batch-Verfahren'. Interaktiv angelegt werden Programme, die wenige oder nur Einmaldaten verarbeiten. Dabei kann über solche 'Zwischenfragen' des Programms auch eine Programmsteuerung selbst vorgenommen werden, etwa bestimmte Unterprogramme aufgerufen werden, wie bei der zunehmend beliebter werdenden 'Menuesteuerung'. 'Im batch' dagegen werden bestehende und 'laufend gepflegte' Kundendateien (etwa auf säumige Zahler, monatlich) durchsucht.

Nun ist die Anlage nach dem Einschalten leer, sie weiß nichts, kann deshalb auch nicht nach dem Namen des Programms fragen, das sie sich einlesen soll. Dies muß man ihr über ein allgemeines Programm sagen, das 'Betriebssystem'. Leider kennt die eben eingeschaltete Anlage auch ihr Betriebssystem nicht, das selbst erst noch geladen werden muß - man muß ihr dies mit einem weiteren vorgeschalteten Programm klarmachen. Dies ist der Urlader (boot strap) als 'Streifen' einst selbst erst vom Band gelesen, heute aber 'resident' und durch eine Taste anzusprechen.

Diese Verallgemeinerung in Stufen ist ein wesentliches Kennzeichen der EDV-Organisation, erst dadurch wird sie flexibel. Man hat so die Wahl, dem Betriebssystem zu sagen, es solle dieses oder jenes Programm in den Arbeitsspeicher laden, das seinerseits dann diese oder jene Benutzerdatei eröffnet und liest. Es gibt aber auch bloße organisatorische Aufgaben, die 'auf der Betriebssystemebene' (oder über ein darüberliegendes Menü), also ohne Anwenderprogramm erledigt werden. Dazu gehört das bloße Ansehen einer Datei, das Löschen einer nicht mehr benutzten Datei, das Umbenennen einer Datei, das Kopieren einer oder mehrerer Dateien von einer Platte auf die andere (Sichern), das Ausdrucken von Dateien usw. Hinter den einzelnen Kommandos für diese Tätigkeiten verbergen sich ganze Programme innerhalb des Betriebssystems - dies ist die Welt der Systemprogrammierer, in die Anwendungsprogrammierer kaum einen Einblick haben.

Die größeren Betriebssysteme haben eine Möglichkeit, solche organisatorischen Befehle, zu denen auch das Starten von Anwenderprogrammen gehört, hintereinander als Ablauf in eine Datei zu schreiben. Man kann diese Datei dann aufrufen, und sie etwa über Nacht 'abarbeiten' lassen. Kann dieser Ablauf von Bedingungen abhängig gemacht werden, etwa Reaktionen auf bestimmte Fehler, so besitzt das Betriebssystem eine 'Prozedursprache'.

Beim Kopieren stellt sich die Frage, ob die Zielplatte die Organisation der Dateien auf der Quellplatte mitübernehmen soll, oder sich eine eigene, günstigere aufbauen darf (die Betriebssysteme der Mikroklasse bieten diese Wahl bisher nicht). Die erste Übertragung nennt sich physische, sie wird bei der bloßen täglichen Sicherung angewandt, weil schneller, die zweite logische, sie dient der 'Reorganisation' der Sammlung. Einzelne Dateien aber können nur in die bestehende Ordnung der Zielplatte aufgenommen, also logisch kopiert werden. Es ist diese von den zugehörigen Programmen unabhängige Plattenorganisation, die es möglich macht, daß ggf. riesige Datenmengen durch einen bloßen Dateinamen verfügbar werden, die aber zugleich verhindert, daß die CPU direkt auf sie zugreift - sie müssen erst von dem jeweiligen Verarbeitungsprogramm an die richtige Stelle im Arbeitsspeicher eingelesen werden. Dabei ergibt sich häufig, daß die zu bearbeitende externe Datei größer ist als der im Arbeitsspeicher verfügbare freie Platz - ein weiterer Grund für die Trennung, das Programm über das Betriebssystem nur gerade das passend große Häppchen an die bestimmte Stelle einlesen zu lassen. War das noch nicht die gewünschte Stelle der Datei, so muß die Suche fortgesetzt werden; das Lesen der Datei liegt typischerweise in einer 'Schleife'.

Die Dateien sind bei den besseren Betriebssystemen in Familien abgelegt. Der Begriff Familie ist genau aber nahezu beliebig weit zu fassen. Interessant dabei sind die Überschneidungen. So wollen in einem Betrieb technische Daten streng getrennt sein von kaufmännischen, die Materiallagerdaten oder Belegungszeiten aber beiden zugänglich; in dem kaufmännischen Bereich (mit den sensibleren Geld-Daten) möchte man Einkauf, Verkauf, Lohn und Gehalt, Kostenrechnung, und Betriebsübersichten voneinander trennen, aber auch Gebiete oder Kundenbereiche der Sachbearbeiter. Dies wären weitere senkrechte Teilungen. Waagerechte Familien entstehen durch Teilung in Stammdateien für Stoffe, Namen, Adressen, Zahlungen, Kosten, Maschinenzeiten, kumulierte Werte usw. Weil auf jede Datei meist viele Verarbeitungsprogramme zugreifen, führen Änderungen in dem Aufbau einer Datei zu der Überarbeitung entsprechend vieler Programme, mit guten Aussichten auf neue Fehler. Die nächsthöhere Kunst über dem Programmieren ist deshalb der schon am Anfang fast richtige Aufbau der Arbeitsdateien.

Ein weiteres Problem, jetzt ein 'mechanisches' des Betriebssystems ist, daß man den verschiedenen Dateien auch noch bestimmte Qualitäten geben möchte, etwa, daß sie nur gelesen, nicht aber verändert werden können, oder daß sie nur von be-

stimmten Plätzen aus oder nur für bestimmte Benutzerschlüssel oder -gruppen änder- oder überhaupt lesbar sind. Es darf auch nicht möglich sein, daß zwei Bearbeiter zur gleichen Zeit an derselben Stelle einer Datei Änderungen vornehmen.

Damit sind wir, ganz nebenbei, auf den 'Mehrbenutzerbetrieb' gekommen, der für die kommerziellen Anlagen eine Selbstverständlichkeit ist. Diese Systeme stellen jedem Benutzer einen Teil des Arbeitsspeichers zur Verfügung, mit denen die CPU in unmerkbar kurzen Abständen kommuniziert. Reicht dieser Platz nicht aus für das für einen der Benutzer tätige Programm, so wird nur immer der Teil von der Platte geladen, der gerade arbeitet, weitere Programmteile aber nach Bedarf in einen Überlagerungsbereich (overlay) nachgeladen. Da bei dem Mehrbenutzerbetrieb Prioritäten vergeben werden, können in solchem Fall die 'Antwortzeiten' des Systems lang werden. Ein weiteres (gelöstes) Problem besteht darin, daß für den Fall, daß zwei Benutzer mit demselben Programm arbeiten, dieses zur Platzersparnis nur einmal im Arbeitsspeicher stehen soll. Die Entwicklung und Verfeinerung der Betriebssysteme auf den gegenwärtigen Stand hat viele Jahre in Anspruch genommen.

Die hier interessierenden Anlagen stehen erst vor der Schwelle der Mehrbenutzerfähigkeit. Die neueren Betriebssysteme erlauben aber immerhin schon ein Abarbeiten zweier 'Tasks'; so kann etwa im 'Vordergrund', auf dem Bildschirm sichtbar, ein Brief geschrieben, im Hintergrund aber, vorher angeworfen, ein Text ausgedruckt werden; 'Multitasking' für entsprechend mehr, bisher wenig wichtig.

Das Betriebssystem ist ein unerläßlicher Bestandteil der Software ('Systemware') und wird vom Hersteller mitgeliefert. Seine Qualität entscheidet meist über die des Rechners. Die Anwenderprogramme, die der Benutzer vom Hersteller oder von Dritten kauft oder sich selbst erstellt, sind immer nur zusammen mit dem zugehörigen Betriebssystem lauffähig.

Wenn also das Wichtigste 'unsichtbar' ist - was sieht der Benutzer dann eigentlich?

Das Harte an der Ware

ist alles Anfaßbare, also auch die Datenträger (Magnetbänder, Platten, ohne die Daten darauf). Nach dem Cheesburgerprinzip gibt es nun auch Firmware, Systemware, und im Kommen ist Knowledgeware.

Anzufassen ist die Tastatur, die nicht mehr fest am BS angeformt sein sollte. Nationale Zeichensätze sind inzwischen üblich, eingeschlossen ß. Der praktische Extrapack für Zahlen liegt leider immer rechts. Zusätzliche Funktionstasten sind sehr nützlich, müssen aber mit dem Betriebssystem abgestimmt sein; nachträglich von anderer Quelle gekaufte bzw. illegal überspielte Programme brauchen ggf. eine etwas knifflige Anpassung. Heutige Tastaturen sind flachkeilig, preisgünstige Modelle zeichnen sich durch wackelnde Tasten mit tiefem 'weichen' Hub aus.

Die übliche Größe des BS ist 12" (30 cm, diagonal), einige Anlagen besitzen 9", Graphikanlagen oft 14" Schirme. Die Diskussion über die richtige Farbe der Darstellung ist schon oft entschieden worden; das hier Gedruckte war grün. Der Rechner selbst kann mit Buchstaben nichts anfangen, er erhält sie von der Tastatur als jeweilige Zahlenkombination, und schickt dieses an den BS, auf dem sie mittelbar nach dem Eingeben erscheint - umgesetzt als Buchstaben, genauer, als Punktmuster. Diese Umsetzung erfolgt nicht im Rechner, sondern im BS. Auf die Fläche des BS sind meist 80 Stellen pro Zeile (weniger ist ungünstig für Textverarbeitung) und 25 Zeilen eingerichtet. Dabei dient die oberste oder unterste Zeile oft als 'Statuszeile' für Darstellung des Betriebszustandes. Komfortabel ist die Möglichkeit, die Stellen einer Zeile zusammenzudrücken, so daß 132 in eine Zeile passen - Jahresübersichten.

Bei graphikfähigem BS kann jeder einzelne Punkt des Stellenmusters angesprochen werden. Die graphische Ausgabe erfreut sich steigender Beliebtheit, Verhältnisse zwischen Zahlen werden in erstaunlichem Ausmaß suggestiv anschaulich. Die farbige Darstellung ist wegen der kleinen Fläche und vergleichsweise geringen Auflösung des BS keineswegs Spielerei. Eine Auflösung von $x:y = 512:256$ kann als befriedigend gelten, professionelle Spitzensysteme haben eine Auflösung von 1024 auf 1024 'pixels'; für 'runde' Kreise ist ein ganzzahliger Faktor zwischen beiden Richtungen günstig. Durchaus sinnvoll ist auch bei Kleinanlagen eine Teilung in einen Text-BS und einen Graphik-BS.

Die Tastatur erlaubt nur die Eingabe in das System, der BS nur die Ausgabe (auch als Widerspiegelung der Eingabe). Ein weiteres Ausgabegerät ist der Drucker. Wie beim BS erfolgt die Ausgabe entweder als fertiges 'Schönschrift'-Druckzeichen (Typenradrunder) oder als Punktmuster (Matrixdrucker). Für Korrespondenz kommt eigentlich nur der Typenradrunder in Frage (möglichst mind. 40 Zeichen/sec.), für bürointerne Darstellung reicht der billigere oder schnellere (bis über 200 Z/s) Matrixdrucker; diese Drucker können neuerdings auch Graphiken ausdrucken, sogar farbig. Eine Neuentwicklung, mit Punktdarstellung, ist der fast unhörbare Tintenstrahldrucker.

Da Texte nach draußen meist die erste große Erleichterung durch die Anlage darstellen, ist ein Typenradrunder zu empfehlen. Dieser müßte aber dann schon die Standardqualitäten heutiger Textdarstellung besitzen, wie Wechsel zwischen 10, 12 und 15 Zeichen/Zoll, anderthalbzeiligen Vorschub, Unterstreichen u.a. Plotter sind Graphik-Ausgabegeräte für Papier und Folie, mit Faserspitze oder teuren Tuschezeicheneinsätzen. Ausführung als Flachbettplotter (DIN A4, A3) oder Trommelplotter; letztere sind im Ingenieurbereich schon verbreitet. Kriterien sind Zeichengeschwindigkeit in der Achse (5 .. 30, Trommelplotter bis über 60 cm/s), Auflösung (0,1 ... 0,03 mm), Wiederholgenauigkeit nach programmgesteuertem Stiftwechsel (möglichst unter 0,3 mm); aber auch die Firmware (eingebaute Programme für Stricharten, Markierungen, Kreise, Achsen, Schriftbehandlung) und die Verwendbarkeit bestimmter Graphikprogramme vom Weltmarkt. Viele Plotter erlauben das Anfahren von Punkten durch Handsteuerung und Übergabe von deren Koordinaten an den Rechner ('Digitalisierung'); dies ist eine echte Dateneingabe.

Eine schnellere Hand-Eingabe erlaubt das graphische Tablett, das ebenfalls Punkte einer Vorlage als x,y Koordinaten digitalisiert. Durch Antippen bestimmter Felder können darüber hinaus Befehle aktiviert oder ein Lichtpunkt ('Cursor') auf dem Bildschirm bewegt werden. Sind diese Felder aber über ein Menü auf dem BS sichtbar, so können sie durch einen 'Lichtgriffel' oder durch Auffahren des Cursors mit der 'Maus', einer Neuentwicklung, für sehr schnelle und eingabefehlerfreie Befehlsfolgen angesprochen werden. Ob sich eine Menüsteuerung mit dem bloßen Finger (HP 150) durchsetzt ist noch offen.

Bleibt die Datenspeicherung. Zugänglich sind nur die Datenträger, die zum Einspielen eingelegt, zur Sicherung nach dem Kopieren entnommen werden, bisher meist Floppys. Übliche Konfiguration sind 2 Floppy-Laufwerke, oder, im Bürobetrieb sinnvoller, eine Floppy und 1 Festplatte. Floppys gibt es in den Größen 8" (inzwischen seltener), 5,25", und 3" .. 3,5", letztere stellen die neueste Entwicklung dar. Bei heute üblicher Aufzeichnung doppelseitig und in doppelter Dichte ('ds, dd') werden bei den verbreiteten 5,25" Disketten 360 .. 1000 Tausend 'Buchstaben' (Bytes, 1000 = 1 kiloByte kB) gespeichert. Festplatten, wie sie bei den großen Textmengen etwa von Ausschreibungen zu empfehlen sind, haben eine Speicherkapazität von 5 .. 40 und mehr Tausend kB (Tausend kB = 1 MegaByte MB).

Die Größe des internen Arbeitsspeichers ist ebenfalls ein wesentliches Kennzeichen der Kapazität. Die bisherige Heimcomputerklasse mit ihrem '8-bit Prozessor' hat max. 64 kB, die zeitgemäßen neuen 16-bit Anlagen haben 128, 256 oder mehr kB. Der Sprung in die größere Speicherklasse ist wegen der leistungsfähigeren Programme (und des Betriebssystems, und des Prozessors) kaum mehr diskutierbar. Was also spielt sich dort ab?

Programme, Leben

Was 'versteh' er eigentlich? Unangenehm, so gut wie nichts. Zählen kann er auch nicht, jedenfalls nicht bis 3. Er addiert 0 oder 1 zu 0 oder 1, Ende. Das Programm ist also dazu da, unsere Wünsche auf diese Ebene von Relaisfunktionen herunterzuübersetzen. Je komplexer wir unsere Wünsche dabei formulieren dürfen, umso mächtiger ist diese 'Programmiersprache'. Dabei gibt es aber nur 3 Arten von Befehlen in einem Programm, arithmetische, logische und organisatorische, in einem Statement z.B.:

Wenn $x * y = 0$ dann mache ...

Für ein kleines Programm, das etwa zu allen Nettopreisen die MWSt addiert, gibt es nun zwei Möglichkeiten der Formulierung, die mathematische, und die umschreibende, mehr oder weniger so:

- $VKPRS = NTPRS + NTPRS * MWST$
 - MULTIPLY NETTOPREIS BY MWST GIVING
 MWBETRAG.
 ADD NETTOPREIS MWBETRAG GIVING VERKAUFS-
 PREIS.

Diesem Befehlsgerüst müßten nun Daten zugeführt werden, viele, denn der Einzelfall würde kaum auf die Maschine gebracht werden. Den Faktor 0.14 der MWSt könnte man als Konstante fest in das Programm schreiben, vielleicht doch nicht praktisch, aber die Nettopreise müßten eingelesen werden. Sie sind auf der Platte in der Artikel-Stammdatei abgelegt. Die Artikel sind vorher einmal über den Bildschirm erfaßt und auf die Platte geschrieben worden, mit Hilfe des (Artikel-) Stammdaten-Verwaltungsprogramms. Die einstige Artikelkartei, bestehend aus einer Karteikarte pro Artikel, ist so zur Artikeldatei geworden, mit 1 'Satz' pro Artikel. Jeder Satz wäre gleich aufgebaut, etwa:

Sa	Art.Nr.	Art.Bez.	Nettoprs
.....

Satzinhalt als Beispiel:

99 001042 Handstr.-zgl,holl..... ...119

Unter Sa, Satzart, könnte man vermerken, ob ein Artikel verfügbar ist oder nicht.

Wollte man sich den Inhalt dieser Artikeldatei Satz für Satz ausdrucken lassen, so geschähe dies sequentiell, wie die Eingabe, weil die Artikel in dieser Reihenfolge in der Datei abgelegt sind. Wünscht man aber irgendeine logische Reihenfolge, etwa aufsteigende Artikelnummern, aufsteigende Preise, alphabetische Reihenfolge der Artikel, ggf. dabei nur einen Ausschnitt, von .. bis, so müßte ein besonderes Programm des Betriebssystemes aufgerufen werden, das mit Hilfe von Zwischendateien eine solche Sortierung vornimmt.

Eine bessere Methode, insbesondere im kommerziellen Bereich üblich, ist die der 'Indexdatei'. Hierbei würde bei Anlegen der Dateistruktur durch das Verwaltungsprogramm ein bestimmtes Feld nach Wahl, oder mehrere, zum Schlüssel (key) erklärt, etwa die Artikelnummer, und dafür automatisch eine kleine Extradatei angelegt, in der die Artikelnummern bei Eingabe alle geordnet eingetragen werden, und zu jeder die zugehörige Satznummer der in beliebiger anderer Reihenfolge eingegebenen Artikelsätze der eigentlichen Datei. Statt später die ganzen echten Sätze sortieren zu müssen, braucht man dann bloß die kleine Indexdatei sequentiell zu lesen, und entsprechend jeder gefundenen Satznummer sich dann die echten Sätze direkt herauszugreifen (automatisch). Dieses System wird in den 'Datenbanken' verwendet.

Unser Verkaufspreis-Programmchen sähe dann so aus (eine Von-Bis Begrenzung ist noch nicht eingebaut):

Abfrage 'welche Datei zu eröffnen?'

XYZ = Antwort

Datei „XYZ“ öffnen

SCHLEIFE.

«nächsten Satz der Datei von der Platte in den Arbeitsspeicher (Pufferbereich) lesen und im Eingangsbereich des Programms bereitstellen; Bei Dateende nach Programmende springen»

IF SA = 99 GO TO SCHLEIFE. «Artikel nicht verfügbar»

$VKPRS = NTPRS + NTPRS * MWST$

«den Verkaufspreis VKPRS mit der Artikelnummer und der Artikelbezeichnung auf einer Liste ausdrucken in einer im Programm bestimmten Formatierung»

GO TO SCHLEIFE.

ENDE.

Datei „XYZ“ schließen.

Die Artikel würden nun entweder einfach in der Reihenfolge, in der sie irgendwann vorher in die Datei geschrieben worden sind, (oder über einen sortiert aufgestellten Index) mit je einem Schleifendurchgang gelesen, verarbeitet, und mit dem Rechnungsergebnis ausgedruckt werden. Jeder Artikelsatz würde auf der Platte gelesen und in den Satzteil kopiert, wo er in die 4 Bestandteile seines logischen Aufbaus aufgegliedert wird. Wir müssen also auf die 35. Stelle zugreifen, da fängt der Preisteil des Satzes an, er ist 6 Stellen lang, die letzten beiden seien im vorliegenden Programm als Pfg. vereinbart, und wir haben

dieses Feld NTPRS genannt. Dieser Name gilt nur in diesem Programm; greift ein anderes, vielleicht von einem anderen Programmierer geschriebenes Programm auf dieselbe Artikel-Stammdatei zu, so könnte dort die in diese Stelle zu lesende Variable anders heißen und, meist ein Fehler, ganz anders aufgefaßt werden, etwa als Geburtstag, oder als Postleitzahl.

Zu bemerken ist noch, daß i.a. Zahlenfelder und Buchstabenfelder unterschiedlich behandelt werden müssen. So werden, wie das Beispiel zeigt, Texte linksbündig abgestellt, Zahlen, wenn mit ihnen gerechnet werden soll, der Stellenlogik wegen rechtsbündig. Ein Problem ist zu ahnen - wie soll das in bloßen 0 1 Ziffern gut gehen?

Gestartet werden könnte unser 'Verkaufspreis-Programm' dann folgendermaßen:

```
RUN 'ARTVKP'  
'welche Datei zu eröffnen?'  
ARTSTM
```

und schon rappelt der Drucker, falls wir daran gedacht haben, ihn vorher einzuschalten.

Nun zu der Frage, was der Rechner eigentlich versteht. Es wird schon aufgefallen sein, daß er * oder MULTIPLY annimmt, obwohl er doch angeblich nur 0 und 1 versteht (wie will er da überhaupt Befehle und Daten, und bei den Daten rechenbare Zahlen von Buchstaben eines Textes unterscheiden?)

Und hier ist der große Stufentrick. Er verarbeitet diese Ziffern 0 und 1 zwar einzeln, 'versteht' sie aber immer nur in Gruppen von 8 oder 16 oder 32 oder 64 Stück. Diese Stellen, die nur mit 0 oder 1 gefüllt werden können, sind die bits. Begrifflich organisiert sind immer 8 bits als 1 byte, dieses könnte demnach 11100010 enthalten, oder 10100011, es gäbe bei 2 Möglichkeiten der Besetzung (0,1) und 8 Stellen genau 2 hoch 8 Varianten, also immerhin 256 Aussagen. Es wird schon sichtbar: sollte er bei Aussage 34 bereit sein zu multiplizieren, so müßten wir ihm einfach sowohl * wie MULTIPLY als '34' schmackhaft machen. Das kleine Extraprogramm, das solches schafft, ist der 'Übersetzer' (Compiler).

Nun stimmt das alles so noch nicht. Der Compiler liest den 'Quellcode', das Programmierer-Programm, löst nach der Syntax seiner Programmiersprache die kompakte 'menschliche' Logik des Ablaufs in eine Vielzahl einfacher Befehle auf, und ordnet diesen die zugehörigen Operanden oder Sprungziele zu, alles in symbolischen Namen. Unsere Kommentare blieben dabei unübersetzt.

Die Multiplikation, * oder MULTIPLY würde damit zu

```
MUL NTPRS,MWST
```

Dieses Geschoß ist die 'Assembler-Ebene', hier trifft man die Systemleute, d.h. nicht, weil man hierhin gar nicht kommt.

Ein weiteres Umwandlungsprogramm, der Assembler, macht dann aus MUL, das als Befehl erkannt wird, weil an erster Stelle, einen Zahlenkode, und hängt diesem die Adressen der zugehörigen Operanden NTPRS und MWST an, als Zahlen. Diese Adressen verweisen auf eine Tabelle, in die der Assembler die Operanden je einmal einschreibt, auch wenn sie mehrfach im Programm angesprochen werden. Weil die Ordnung durch erkennbare Symbole, die der Compiler noch eingehalten hat, bei Umsetzung in bloße Zahlen verloren geht, muß sie durch eine vereinbarte Stellung dieser Zahlen wiederhergestellt werden. Der Assembler löst außerdem alle Zahlen und Buchstaben, d.h. Byte-Zeichen in Bitkombinationen des 'Maschinenkodes' auf, diese sind wegen 8:1 auf dem BS nicht mehr darstellbar.

Der Programmablauf bestünde nun darin, daß die Befehle der Reihe nach abgearbeitet würden, wobei die zugehörigen Operanden über ihre Adressen zugänglich sind.

Leider ist das noch nicht die ganze Wahrheit. Der Raum, in den der Assembler seine Übersetzung in Zahlen, das 'Objekt' einschreibt, und die zugehörigen Operanden- und Sprungadressen berechnet, ist nur ein Teil des Arbeitsspeichers, darunter und darüber liegen noch Bereiche, in denen das Betriebssystem angesiedelt ist, und in die unser Programm keinesfalls irgendwo hineinragen darf. Der Assembler berechnet alle seine Adressen nur ab Christi Geburt, d.h. relativ; lauffähig aber ist das Programm erst mit absoluten Adressen.

Der gute Grund für diesen Mangel ist, daß die Teile eines Programms verschieblich bleiben sollen, um in ganz verschiedene Gesamtprogramme eingebunden werden zu können. Man

muß deshalb noch zu allen Adressen des Moduls im jeweiligen lauffähigen Programm die Distanz bis zum wirklichen Anfang des Arbeitsspeichers hinzuzählen. Dies besorgt ein weiteres Hilfsprogramm, der Binder (Linker); aus dem verschieblichen Modul (relocatable object) wird so das lauffähige (run-time module).

Zum Abschluß noch ein zu kurzer Hinweis auf die Verschiedenheit von Zahlen und Buchstabenkode. Zahlen werden, um rechnerisch verarbeitet werden zu können, in Exponential-schreibweise abgestellt, Buchstaben eines Textes aber in den amerikanischen ASCII-Kode umgesetzt. Die Logik, zwischen „Ich biete 4,5 fordert er“ und 4,5 als Zahl

zu unterscheiden, ist Sache des Compilers (der vorstehende Handel fände aber in einem Textverarbeitungsprogramm statt, das sowieso bisher noch nicht rechnen kann).

Das Einziehen der symbolsprachlichen Ebene und der des verschieblichen Objektes waren große Fortschritte. Die Wahrheit ist allerdings, daß die Entwicklung andersherum, von unten nach oben verlaufen ist. Die kaum überschätzbare Bedeutung dieser scheinbaren Verkomplizierung aber liegt nicht einfach darin, daß der Rechner in seiner Maschinensprache, und der Programmierer in seiner problemorientierten Darstellung arbeiten kann. Es ist vielmehr nun möglich, daß immer neue, leistungsfähigere Rechner entwickelt werden - die vorhandenen Programme aber übernommen werden können, weil in maschinenunabhängigen Hochsprachen geschrieben. Es braucht bei jedem Fortschritt nur der Assembler geändert zu werden, in besonderen Fällen auch der Compiler, und dann werden die alten Programme einmal neu übersetzt und gebunden, schon läuft die gute alte Organisation auf der besseren neuen Anlage.

Schließlich aber sind nicht nur diese Programme inzwischen ein Multimilliardenbesitz der Zivilisationen, sondern auch die Fähigkeit vieler tausender von Anwendungsprogrammierern, ihre gelernten Sprachen über Jahre hinweg einsetzen zu können.

Sprachen, Schönheit

Wir betreten das Softwarehaus im EG, Betriebssystemebene, sehen uns den Keller (Assembler, Treiberroutinen u.a.) gar nicht an, auch die interessanten Auslagen im EG nicht (leider, nämlich Textverarbeitung, Tabellenprogramme), Datenbanksysteme, Graphik), sondern fahren in das OG.

Hier gibt es die klare Teilung in kommerzielle und technisch-wissenschaftliche Sprachen. Aber nicht bei allen ist diese Teilung ganz klar, wie auch Großfirmen eigene Forschung betreiben, und sich dann nicht gerne zwei verschiedene Stäbe von Programmierern halten.

Die weitestverbreitete kommerzielle Sprache ist COBOL, mit weiträumiger Dateienverwaltung, deutlichen Anfängen einer Strukturierung, hervorragenden Möglichkeiten der Druckbildgestaltung, eigenem Datenformat, aber piepsigen 4 Grundrechenarten, viel zu langsam, kaum netzwerkfähig, und eben etwas veraltet inzwischen. Konrad Zuse, der deutsche Pionier der EDV, dessen welterster programmierbarer Rechner 1943 lief, soll bei Erscheinen dieser Sprache sein Hersfelder Werk aufgegeben haben. Cobol spielt im Microbereich keine tragende Rolle.

Ganz abgelebt ist RPG-II, der interessante Versuch von IBM, noch ein Stockwerk über den Hochsprachen einen nur ergebnisbeschreibenden Formalismus einzuführen.

In der technischen Abteilung stoßen wir zunächst auf FORTRAN, ein prähistorisches Biest, wie manche sagen, die dann immer noch weiter damit arbeiten. Fortran ist für Kleinrechner gar nicht so schlecht, weil es in kompakten und sehr schnellen Objektcode umwandelt - Enge und Langsamkeit dieser Klasse sind ja der Preis, für diesen Preis. Zudem enthält das neue Fortran-77 einfache und wichtige Strukturverbesserungen. Es fällt auch einem Systemmann nicht allzu schwer, ein Verbindungsprogramm für Ausgabe auf einen Graphik-BS zu schreiben, oder ein anderes, das es erlaubt, die einzelnen Stellen des Textbildschirms als Koordination anzusprechen, so daß echte und schnelle Masken möglich werden.

Damit macht Fortran schon mehr Freude.

Als Dateien sind, einfache Handhabung, sequentielle und direkte möglich, Indexdateien müßten im eigenen Programm verwaltet werden.

Gleich auf den nächsten Tischen schon liegen die vielen BASIC's, Derivate von Fortran, Lieblingssprache der kleinen

Leute (6-16), aber immerhin von einer Firma wie HP zu einer leistungsstarken Schnellentwicklungssprache für Labor und Technik erweitert. Basic war als Schulsprache konzipiert worden, jeder Fehler (in der Syntax, nicht in der Logik!) sollte sofort angezeigt und -interaktiv- verbessert werden können. Dies bedeutet, daß der Übersetzer dauernd im Speicher steht, und jede Zeile sofort einzeln übersetzt und dabei auf Richtigkeit prüft. Man nennt diese Familie Interpretersprachen, im Gegensatz zu den schon beschriebenen Compilersprachen. Wegen der zeilenweisen Übersetzung, bei jedem Lauf neu, sind sie langsamer. Manche Versionen allerdings können im nachhinein kompiliert werden (wie die populäre von Microsoft), und sind dann so schnell wie Fortran. Programmierer haben weniger Schwierigkeiten mit der Syntax, und schätzen an den verschiedenen Möglichkeiten des Kompilierens auch den Vorzug, den Quellcode mit einem komfortableren Editor schreiben zu können, wie einen Brief; sie brauchen Basic weniger.

Um zu zeigen, wie leicht ein kleiner Zeittest aufzustellen ist, hier ein 'benchmark' Programm. Es beginnt mit einer Frage nach der Anzahl der Schleifen, die durchlaufen werden sollen. Gibt man etwa die Zahl 10000 ein, so multipliziert der Rechner einen fest vorgegebenen Wert mit sich selbst, er tut dies 10000 mal. Ist er damit fertig, gibt er das Ergebnis der einfachen Multiplikation einmal aus:

Auf einer 8-bit CP/M Anlage hat dieses Programm dann 67 Sek. gebraucht, um die gegebene 10-stellige Zahl mit 11-stelliger Richtigkeit 10000 mal zu bearbeiten. 1 Multiplikation (und 1 Rücksprung, sowie Hochzählen der Schleife) hat damit 6,7 Millisek. gedauert. In der Compilersprache Fortran war das gleiche 'Problem' bei einfacher Genauigkeit in 13 sec, bei doppelter in 51 sec gelöst. Großrechner brauchen weniger als 1/1000 dieser Zeit.

Mit PL/1 hat IBM für Großrechner eine Sprache geschaffen, welche die Fähigkeiten und Formalien von Cobol und Fortran vereinigt und erweitert. Ein Micro-Subset davon ist PL/M.

ALGOL war die erste Sprache, die einen anspruchsvollen Aufbau hatte, und man war begeistert, als der einfachere Nachfolger da war, PASCAL, die einzige bedeutende nicht in den USA entwickelte Sprache. Sie ist allerdings in der Ein- und Ausgabe nicht eben groß entwickelt, und *Wirth*, ihr Schweizer Schöpfer, hat gerade sein neues letztes Wort vorgestellt, MODULA-2. Eine andere Weiterentwicklung und Vergrößerung von Pascal ist die neue Sprache ADA, wie einst Cobol mit Unterstützung des Pentagons entwickelt.

Eine weitere zunehmend beliebte Sprache ist C, kompakt und schnell fast wie ein Assembler, jedoch mit komplexer wirkenden Befehlen und Strukturen. Viele Compiler anderer Sprachen werden selbst in C geschrieben.

Das Neue an der Algol-Familie, wie man wohl bald nicht mehr sagen darf, ist der starke Anbau logischer, d.h. nicht-numerischer Fähigkeiten. Das entspricht immer weiter sich aufbauenden Wünschen. Nicht nur stereotype, wenn auch komplizierte mathematische Dienstleistung wird verlangt, sondern auch sich möglichst selbst steuernde Kombinationsfähigkeit. Die bisherige EDV verarbeitet Daten zu Informationen, die kommende der 5. Generation soll über Verfahren der künstlichen Intelligenz Informationen zu Wissen verarbeiten (Knowledgeware).

Auf diesem Weg liegt der Aufbau von Datenklassen, die mehr oder weniger miteinander verwandt sind, um die Strukturierung in viele als Funktionen aufzurufende Module; das schlichte GO TO ist eliminiert worden. Um die Sprachen trotzdem einfach und hantierbar zu halten - zu der Gesamteffektivität eines Programms gehört auch der Aufwand der Erstellung, und zwar umso mehr, je weniger oft das Programm eingesetzt wird - mußten sie noch weiter von allem Handwerklichen getrennt werden. Zwei Sprachen haben dies schon vor langem getan, Zukunftssprachen bis heute.

Die eine dieser beiden ist APL; Stenographie wäre eine etwas unfreundliche Bezeichnung für dieses intellektuelle Vergnügen. Wer auf leichte Art in den gedanklichen Kreis der EDV eindringen möchte, dem kann die angenehme Einführung von *Cilman* und *Rose* in APL als Bettelkürze empfohlen werden. APL besitzt einen eigenen Zeichensatz, den man lernen muß (so leicht und natürlich wie russisch, wenn man sich wirklich daranmacht). Die Sprache arbeitet immer mit Zahlenreihen oder ganzen Feldern, auf vielen Ebenen, und führt zu einem neuen Denken.

Interaktiv wie APL ist auch LISP, ein Dornröschen hinter Klammern, und ein solches Potential, daß die derzeitige Administration die Ausfuhr ihrer leistungsfähigsten Versionen aus den Staaten zu verhindern sucht. Es hat nicht an Bemühungen gefehlt, beide zu zähmen, ohne wesentliches ihrer Fähigkeiten aufzugeben. Bekannter ist die Vereinfachung von Lisp geworden, LOGO.

Schließlich ist noch eine weitere Expertensprache zu erwähnen, FORTH. Der Benutzer kann sich beliebige Funktionen bilden, diese in weitere Funktionen einbauen (und sich dabei selbst aufrufen lassen - Rekursion, das kann schon Pascal), so daß überaus kleine und mächtige Module aufbaubar sind.

Und nun stellt sich die Frage, mit welcher Sprache wir jetzt an die Kasse gehen. Noch vor wenigen Jahren wäre das Basic gewesen, für Anfänger das einfachste. Mit den neuen Möglichkeiten aber beginnt man auch über den Bildungswert der Erstsprache nachzudenken und findet, daß die einfache Kontrollstruktur insbesondere von Basic aus den Programmen eine Art von Spaghetti macht. Na, und der Koch?

Wichtiger als die Entscheidung in der Sprache (die dritte lernt man sowieso leichter als die zweite) ist die Frage, wie man an das Lernen herangeht, um ständig kleine Erfolge und Ermutigungen zu erreichen. Damit ist die letzte Frage eigentlich schon beantwortet, soll man programmieren lernen oder einfach nur fertige, gute und leistungsfähige Programme einsetzen?

Fertige Programme liefern fertige Ergebnisse. Kann man sie brauchen, ist man fertig. Aber daß die Probleme des immer herkömmlicheren Bauens nach der EDV im weiteren Sinne riefen, ist nicht zu hören (die gesamte Textverarbeitung ausgenommen; wer einmal vor dem Bildschirm saß -). Es stellt sich die Frage, ob der Käufermarkt den nachwachsenden Architekten noch die Möglichkeit bietet, das bisherige Selbstverständnis ihres Berufes fortzuführen. Man kann deutlicher fragen - wohin werden die Begabten sich in Zukunft wenden?

Was nun?

(1) nicht kaufen.

Es trifft zu, daß die Preise weiter fallen, und die Leistungen ganz erstaunlich steigen, eine wohl einzigartige Marktlage. Dabei stellen diese Leistungen einen Zusammenhang dar:

- Der Arbeitsspeicher wird größer und schneller. Er wird z.T. schon so groß, daß in abgetrennten Bereichen ein externer Speicher nachgebildet wird; damit müssen häufig gebrauchte Dateien (Editor, Übersetzer und Binder, Druckprogramm, die aktuelle Arbeitsdatei) gar nicht erst von der Platte gelesen oder auf sie hinausgeschrieben werden, man hat sie vielmehr schon im Haus, in einem anderen Stockwerk ('RAM-Disk', früher Pseudo-Floppy genannt. Erf. mind. 512 kB; nicht jedes Betriebssystem geeignet).
- Die kleinen Festplatten sind eine Neuheit, die den Micro endgültig zur Bedrohung der bisher erfolgreichen kommerziell schon nutzbaren Minis haben werden lassen. Die Leistungen von 60 TDM Speichern werden heute von 4 TDM Platten eingestellt; man kann sich kaum vorstellen, welche Schockwellen das schlägt. Der Nachteil der Festplatte ist, daß sie nur ein relativ externer Speicher ist; zum Überspielen von Herstellerprogrammen (etwa bei der Erstinstallation) braucht man immer noch zusätzlich ein Floppy- oder ein Magnetbandlaufwerk. Ein Speicherbereich im Arbeitsspeicher (RAM-Disk) erlaubt nun das Spar-Konzept, ohne Festplatte, nur mit 1 Floppy-laufwerk auszukommen.
- Der Preisverfall ist dabei, die Firmen zur Spezialisierung ihrer Arbeit und zur 'Standardisierung' ihrer EDV-Erzeugnisse zu zwingen; die Prognosen darüber, welche Firmen diese Entwicklung überstehen wechseln. Die Vertriebsformen sind ebenfalls noch im Fluß, die Beratung ist meist unzureichend.
- Aber für einen großen Markt mehr oder weniger kompatibler Anlagen lohnt sich jetzt ein hoher Software-Entwicklungsaufwand. Der Massenverkauf zu geringeren Preisen mildert dabei ein bisheriges Problem der Micro-Klasse, den Software-Klau.

- Die so entstehende neue Software ist nicht nur erschwinglicher, und durch die neue Vergleichbarkeit auch besser und sicherer, sie bildet auch neue Ideen ab, oder macht Entwicklungen aus der Groß-EDV zugänglich (die hier, komfortabel gemacht, noch einmal vermarktet werden). Fast jeden Monat gibt es Neues.

Ergebnis:

Verbilligung und Verbesserung gehen schubweise, aber der Trend ist anhaltend. Wartete man also, hätte man die Chance, sicherer und mit weniger Geld in einer neuen Leistungsklasse zu beginnen.

Es gibt ebenso ernstzunehmende Gründe, mit der Anschaffung nicht zu warten:

- Zum einen sind die Anschaffungskosten für jeden ein sehr verschiedener Wert. Ist die gewünschte Leistungsklasse bereits auf dem Markt, so muß man fragen, ob die zu erwartenden Preissenkungen die Wartezeit aufwiegen. In der normalen kommerziellen Praxis ist diese Frage keine mehr.

Sinnvoll für kleine Büros kann es sein, ein billiges Einplatzgerät zu beschaffen, mit einem guten Drucker. Kleingeräte sind nicht bei jedem Finanzamt absetzbar, aber eine Miete. Im Mietvertrag den Service einbauen, vor allem bei abgeleasstem Gerät. Im Büro müßte jetzt ein Engagierter am Gerät Kompetenz erwerben; zugleich könnten Texte schon über den Drucker laufen - es gibt kein zurück zum alten Hammerwerk.

Damit wäre die richtige Anschaffung noch verschoben, man hätte aber wichtige Zeit genützt.

- Der Einplatzmann/frau, der ein EDV-Gerät beruflich nutzen möchte, seine Möglichkeiten und vor allem seine Ziele aber noch nicht recht abschätzen kann und deshalb vor hat, sich am Gerät zu qualifizieren, wird überrascht sein, daß er dazu viel Zeit braucht; und „es ist bereits später ...“.

(2) *Sich erst 'schlau machen'*

Der kürzeste Weg, zugleich der zweitsteilste ist, sich aus der unübersichtbaren Zeitschriftenliteratur Kennzeichen zusammenzutragen, zu Anlagen, Laufwerken, Druckern, Plottern, allg. und Fachsoftware, jeweils mit den Firmennamen verbunden, mit Preisen. Man wird erstaunt sein, was an Firmen in schneller Folge kommt und plötzlich fehlt. Schon geringe engl. Sprachkenntnisse werden ziemlich bald dahin führen, nur noch am Zeitschriften zu lesen.

An diese Fachworte lagern sich dann bald Erklärungen aus Fachaufsätzen an, und damit wird man gesprächsfähig.

Es gibt verschiedene Taktiken der Verkäufer, Leuten mit der Schlaumachabsicht zu begegnen. Allein, wenn irgendwo, dann gilt hier das Prinzip Hoffnung: Man weiß eben auch, daß der unwissende Interessent zugleich der dankbare ist. Und dann tut es dem in seiner Firma von den Programmierern wegen seiner 'gänzlichen Ahnungslosigkeit' geschmähten Vertriebsmann auch einmal wohl, sein EDV-Wissen ungehemmt abzustrahlen. Man benutze Ausstellungen.

(3) *Die subjektive Seite kennenlernen*

Der schnelle Lustkauf führt wegen der Anlaufzeit meist zur Enttäuschung, das nachträgliche Kennenlernen von Kriterien und Bedarf zu dauerhafter Ablehnung.

Ist man aus dem gefährlichen Stadium heraus, Gretchen nur vor dem Dom gesehen zu haben, wird man wählerischer. Allein, es ist immer noch das, was man umarmen kann, hardware.

Das muß auch noch weg.

Nach längerem Gebrauch, später, wird die Anlage sowieso unsichtbar, vorausgesetzt sie läuft störungsfrei. Man arbeitet dann nicht mehr mit ihr, sondern mit dem Programm. Die Seele entscheidet.

(4) *Das Betriebssystem*

Mit dem Betriebssystem ist auch die Familie bestimmt, also der Prozessor, und 1- oder Mehrplatzsystem; das Angebot an interessanten Sprachen und Programmen hängt außerdem von dem eigenen Diskettenformat ab, aber Händler können meist konvertieren.

8-bit Systeme (Z-80) wären technisch für den Einplatz-Anfänger mehr als ausreichend, nur wird hier schon jetzt die Software nicht mehr weiterentwickelt. Die nächste, entscheidend größere 16-bit Klasse, Intel 8088 und 8086, ggf. mit Fließkomma-Prozessor, ist weniger als entsprechend teurer. Ist man aber schon auf dem Wachstumspfad, wird man sich fragen, ob es dann nicht gleich der Motorola 68000 sein soll, oder der neue 80286.

Das Betriebssystem der 8-bit Klasse, CP/M, ist einfach und bewährt, es steht etwas erweitert nun auch für die 16-bit Klasse zur Verfügung (CP/M-86, CP/M-68K); ein zeilenorientiertes Textschreibprogramm ('Editor') ist nicht mehr zu akzeptieren, man verlange ein bildschirmorientiertes, mit freiem Cursor. Etwas leistungsfähiger (Multitasking, Dateien- und Satzschutz) ist das neue Concurrent CP/M 86, das zum Mehrplatzsystem erweitert werden soll.

In scharfem Wettbewerb mit diesen Betriebssystemen der Firma Digital Research steht das MS-DOS von Microsoft (auf IBM als PC-DOS implementiert). Ein wesentlicher Vorzug dieses Systems ist die Baumstruktur des Dateizugriffs (und der entsprechenden Auswahl-Anzeige des Inhaltsverzeichnis großer Festplatten).

Ein interessantes, ursprünglich für den Unterricht entwickeltes Betriebssystem ist das UCSD-p System mit der Fähigkeit, sehr schnell Programme in Pascal oder Fortran und Basic fehlerfrei zu machen. Es wird daran gearbeitet, den verwaltbaren Speicher der heutigen Größenentwicklung anzupassen.

UNIX (zugehörig XENIX, CROMIX, ONIX), das neue Zauberwort der Branche, wird als schwierig bezeichnet, und scheidet als Anfangssystem mit Sicherheit aus. Sehr gelobte Systeme aus dem kommerziellen Bereich sind OASIS und BOS.

(5) *Wo kaufen?*

Der Händler wird nach dem Kauf noch einige Ratschläge geben müssen. Er freut sich nicht darüber, weiß aber, daß der zufriedene Kunde gerade in der EDV das Wichtigste ist. Ein überregional tätiger Vertrieb ist von solchen Empfehlungen weniger abhängig, und kann späteres Fragen als zudringlich empfinden. Nicht bei der ersten, aber bei der zweiten Vorführung nehme man einen schon erfahreneren Beobachter mit.

Der Händler sollte mehr als einen zufriedenen Kunden nennen können, der Erfahrungsaustausch mit den kommenden Kollegen ist notwendig. Der Händler sollte auch Verbindung mit einem Systemmann haben; nachträgliche Ergänzungen etwa als Plotter, Digitalisiergerät, Graphikdrucker bedürfen manchmal der Anpassung. Der Zusammenkauf der Anlage bei mehreren Händlern (aus Kostengründen?) ist ein Fehler, an den man noch lange denken wird.

Ein größeres Büro sollte für bloße Anwendung eine Festplatten-Anlage von einem Großhersteller beziehen, direkt von dessen Niederlassung. Billiger geht es nicht. Ein großer Anbieter hat nicht nur eine genaue Übersicht über die verfügbare und garantiert lauffähige Software - die Entscheidung liegt hier - sondern repariert als Vertragspartner auch alles selbst, und zwar kurzfristig. Ein Händler dagegen nennt einem, nach Ablauf der höchstens 1-jährigen Garantie, meist nur jemanden, der das Laufwerk dann mal justiert o.ä. Meist kennt man dann selbst schon einen Fachmann. Zum Glück sind die meisten Geräte inzwischen robust und über Jahre zuverlässig.