

# Echte Programmierer meiden PASCAL

Vor langer Zeit, in der goldenen Ära der Computer, war es noch einfach, die Männer von den Memmen zu trennen (mitunter auch "Echte Männer" und "Müslifresser" genannt). Echte Männer programmierten Computer und Müslifresser ließen es bleiben. Ein echter Computerprogrammierer sagte Dinge wie "DO 10 I=1,10" oder "ABFID" und der Rest der Welt quängelte "Computer sind mir zu kompliziert" oder "Ich kann zu Computern keine gefühlsmäßige Bindung aufbauen - sie sind zu unpersönlich". Dabei zeigt schon Remy Eyssens Buch "Echte Männer mögen kein Müsli" (Heyne TB 6290), daß Echte Männer zu nichts und niemandem eine "gefühlsmäßige Bindung" aufbauen, und daß sie auch keine Angst haben, unpersönlich zu sein.

Aber die Zeiten ändern sich. Heute stehen wir einer Welt gegenüber, in der kleine alte Damen vollcomputerisierte Mikrowellenherde kaufen können, in der 12 Jahre alte Dreikäsehochs gestandene Männer bei ASTROIDS und PACMAN sattmachen, und in der jeder seinen eigenen Homecomputer kaufen und sogar verstehen kann. Der Echte Programmierer ist gefährdet, von Studenten mit einem igITT 2020 (deutsche Version des ABFALL-II, Anm. d. Übers.) im Gepäck ersetzt zu werden!

Es gibt allerdings einige Unterschiede zwischen dem typischen, PACMAN-spielenden Gymnasiasten und einem Echten Programmierer. Die Kenntnis dieser Unterschiede wird den Heranwachsenden ein Ziel geben, nach dem sie streben können - ein Vorbild, eine Vaterfigur. Außerdem schützt sie den Echten Programmierer vor Arbeitslosigkeit.

Der einfachste Weg, einen Echten Programmierer zu erkennen, führt über die von ihm benutzte Programmiersprache. Echte Programmierer benutzen FORTRAN. Müslifresser benutzen PASCAL. Niklaus Wirth, der Schöpfer von PASCAL, wurde einmal gefragt, wie man seinen Namen ausspreche. "You can either call me by name, pronouncing it Veert, or call me by value, worth", antwortete er. Diese Bemerkung zeigt sofort, daß Wirth ein Müslifresser ist. Der einzige Parameter-Übergabe-Mechanismus, den Echte Programmierer akzeptieren, ist call-by-value-return (call-by-result, Anm. d. Übers.), wie er in den IBM/370 FORTRAN-G- und -H-Compilern implementiert ist. Echte Programmierer brauchen schließlich keine abstrakten Konzepte, um ihre Arbeit zu erledigen; sie sind vollkommen glücklich mit einem Lochkarten-Stanzer, einem FORTRAN-IV-Compiler und einem Bier. Echte Programmierer erledigen Listenverarbeitung, Zeichenketten-Manipulation, Graphikdarstellungen (wenn überhaupt) und künstliche Intelligenz in FORTRAN. Was sie mit FORTRAN nicht machen können, machen sie in Assembler, was sie in Assembler nicht machen können, lassen sie verächtlich liegen.

Akademische Computerspezialisten sind in den letzten Jahren auf's Abstellgleis der strukturierten Programmierung geraten. Sie behaupten, daß Programme verständlicher werden, wenn bestimmte Sprachkonstrukte und Programmtechniken benutzt werden. Sie können sich natürlich nicht einigen, welche Konstrukte am besten geeignet sind, und die Beispiele, an denen sie ihren speziellen Standpunkt aufzeigen wollen, passen ausnahmslos auf eine Seite irgend eines obskuren Journals.

Als ich aus der Schule kam, dachte ich, ich sei der beste Programmierer der Welt. Ich konnte ein unschlagbares TIC-TAC-TOE-Spiel (Vier-in-einer-Reihe, Anm. d. Übers.) schreiben, beherrschte 5 verschiedene Programmiersprachen und schrieb fehlerfreie 1000-Zeilen-Programme. Dann kam ich in die Wirklichkeit. Meine erste Aufgabe bestand darin, ein 200.000-Zeilen-FORTRAN-Programm zu lesen, zu verstehen und um den Faktor 2 zu beschleunigen. Jeder Echte Programmierer wird einem versichern, daß die gesamte strukturierte Programmierung der Welt in einem solchen Fall nicht hilft - hier braucht man wirklich Talent.

Einige Beobachtungen zum Thema "Echte Programmierer und Strukturierte Programmierung":

- Echte Programmierer haben keine Angst vor GOTO's.
- Echte Programmierer schreiben 5 Seiten lange DO-Schleifen, ohne durcheinander zu geraten.
- Echte Programmierer lieben das arithmetische IF-Statement (das mit den 3 Ausgängen, Anm. d. Übers.), weil es den Code interessanter macht.
- Echte Programmierer schreiben selbstmodifizierende Programme, speziell, wenn sie damit in einer kleinen Schleife 20 ns einsparen können.
- Echte Programmierer brauchen keine Kommentare; das Programm ist schließlich selbstdokumentierend.
- Da FORTRAN strukturierte IF, REPEAT UNTIL oder CASE Anweisungen nicht kennt, braucht sich der Echte Programmierer nicht sorgen, daß er sie nicht benutzt. Außerdem kann man sie nötigenfalls über "ASSIGNED GOTO's" simulieren.

Datenstrukturen sind in letzter Zeit in gewissen Kreisen populär geworden. Wirth, der Müslifresser, verfaßte sogar ein ganzes Buch ("Algorithmen und Datenstrukturen", Teubner-Verlag), in dem er behauptet, daß man Programme schreiben könne, die auf Datenstrukturen aufbauen, statt es umgekehrt zu machen. Wie jeder Echte Programmierer weiß, gibt es nur eine wirklich nützliche Datenstruktur, das Array. Zeichenketten, Listen, Records und Mengen sind allesamt Sonderfälle von Arrays und können auch so behandelt werden, ohne dadurch die Sprache zu verkomplizieren. Das Schlimmste an den ganzen schönen Typen ist, daß man sie deklarieren muß, während Echte Programmiersprachen, wie man weiß, den Typ anhand des ersten Buchstabens eines maximal 6 Zeichen langen Bezeichners implizit festlegen.

Welches Betriebssystem der Echte Programmierer benutzt? CP/M? Gott bewahre! Das ist doch im Grunde ein Spielzeug-Betriebssystem. Selbst kleine alte Damen und Hauptschüler können CP/M benutzen und verstehen.

UNIX ist natürlich schon viel komplizierter - der typische UNIX-Hacker weiß nie, wie das PRINT-Kommando dieser Woche heißt - aber wenn man es genau nimmt, ist UNIX auch nur ein verherrlichtes Telespiel. Niemand arbeitet auf UNIX-Systemen an ernstzunehmenden Dingen - man schickt kleine Witzchen über USENET rund um die Welt, oder man schreibt ein neues Adventure-Spiel oder Forschungsbericht.

Nein, der Echte Programmierer benutzt OS/370. Ein guter Programmierer kann die Beschreibung des Fehlers IJK3051 in seinem JCL-Manual (JCL=Job Control Language, Batch-Kommandosprache, Anm. d. Übers.) finden und interpretieren. Ein sehr guter Programmierer erkennt die Fehler, ohne in sein Manual zu sehen. Ein wahrhaft außerordentlicher Programmierer kann Fehler in einem 6-Megabyte-Hexdump finden, ohne hierfür einen Taschenrechner zu benutzen.

OS/370 ist ein wirklich bemerkenswertes Betriebssystem. Mit einem einzigen falsch platziertem Leerzeichen kann man die Arbeit mehrerer Tage zerstören, was die Wachsamkeit im Programmiererteam ungemein fördert. Der beste Weg zum System ist der Kartenstanzer. Zwar behaupten einige Leute, es gäbe ein Timesharing-System unter OS/370, aber nach sorgfältigen Nachforschungen bin ich zu dem Schluß gekommen, daß sie sich irren.

Welche Werkzeuge der Echte Programmierer benutzt? Nun, theoretisch könnte er seine Programme über die Maschinenkonsole eingeben und laufen lassen. In den frühen Tagen der Computerei, als Computer noch Maschinenkonsolen hatten, wurde dies auch gelegentlich getan. Der typische Programmierer wußte den System-Urlader Bit für Bit auswendig und tastete ihn ein, sobald er von seinem Programm zerstört worden war. Damals war Speicher auch noch Speicher - der war nicht einfach leer, wenn der Strom ausfiel.

Hauptspeicher von heute vergessen entweder Dinge, die sie behalten sollen, oder halten Informationen, die schon lange weg sein sollten. Aber zurück zum Thema. Die Legende besagt, daß Seymour Cray, der Erfinder des Cray-I-Supercomputers und der meisten anderen Rechner von Control Data selbst das erste Betriebssystem für die CDC 7600 an der Maschinenkonsole eingetastet hat, als sie das erste Mal eingeschaltet wurde. Cray ist selbstverständlich ein Echter Programmierer.

Einer der Echten Programmierer, die ich am meisten bewundere, arbeitet als Systemprogrammierer für Texas Instruments. Eines Tages erhielt er ein Ferngespräch von einem Benutzer, dessen System mitten in einer wichtigen Arbeit abgestürzt war. Der Typ reparierte dann den Schaden über's Telefon. Er brachte den Benutzer dazu, Systemtabellen in Hexadezimal zu reparieren und Registerinhalte über's Telefon durchzugeben. Die Moral von der Geschichte: obwohl ein Echter Programmierer normalerweise Kartenlocher und Schnelldrucker benutzt, kommt er im Notfall auch mit Maschinenkonsole und Telefon aus.

In einigen Firmen besteht die Programmeingabe allerdings nicht nur aus 10 schlangestehenden Ingenieuren, die auf einen 029-Locher warten. In meiner Firma z.B. steht kein einziger Kartenstanzer. Der Echte Programmierer muß in diesem Falle seine Arbeit mit einem Text-Editor erledigen. Auf den meisten Rechnern stehen verschiedene Editoren zur Verfügung, und der Echte Programmierer muß aufpassen, daß er einen erwischt, der seinen persönlichen Stil wiedergibt. Viele Leute glauben, daß die besten Editoren der Welt am Xerox Palo Alto Research Center geschrieben werden und auf Alto- oder Dorado-Computern laufen. Unglücklicherweise würde jedoch kein Echter Programmierer einen Computer mit einem Betriebssystem benutzen, das Smalltalk heißt, und sicherlich auch nicht über eine Maus mit einem Rechner kommunizieren.

Einige Konzepte der Xerox-Editoren sind mittlerweile in Editoren eingeflossen, die unter sinnvoller benannten Betriebssystemen arbeiten, so wie EMACS oder VI. Das Problem mit diesen Editoren ist, daß Echte Programmierer das Konzept "Du kriegst, was Du siehst" für schlecht halten. Der Echte Programmierer will einen "Du hast es so gewollt, da hast Du's"-Editor, einen, der kompliziert ist, skeptisch, leistungsfähig, gnadenlos und gefährlich. TECO, um genau zu sein.

So wurde beobachtet, daß TECO-Kommandofolgen dem Leitungsrauschen ähnlicher sind als lesbarem Text. Eins der unterhaltsameren Spiele, die mit TECO möglich sind, besteht darin, den eigenen Namen als Kommando einzugeben und zu raten, was dann passiert. So ungefähr jeder mögliche Tippfehler kann das gerade editierte Programm zerstören, oder schlimmer noch, kann kleine mysteriöse Fehler in einstmals funktionierende Unterprogramme einbringen.

Aus diesem Grunde editieren Echte Programmierer nur sehr widerwillig Programme, die schon fast laufen. Sie finden es viel einfacher, den binären Opcode direkt zu ändern, für gewöhnlich mit einem wundervollen Programm, das SUPERZAP heißt (auf Nicht-IBM-Rechnern entsprechend anders). Dies funktioniert so gut, daß viele laufende Programme auf IBM-Systemen keine Ähnlichkeit mit den ursprünglichen FORTRAN-Quellprogrammen haben. In einigen Fällen ist nicht einmal mehr das ursprüngliche Quellprogramm vorhanden. Wenn dann der Zeitpunkt gekommen ist, so ein Programm zu ändern, würde kein Manager auch nur daran denken, einem geringeren als einem Echten Programmierer diese Arbeit zu übertragen - kein müslifressender strukturierter Programmierer wüßte auch nur, wo er mit der Arbeit anfangen soll. Man nennt das Arbeitssicherungsmaßnahme.

- Hier eine Liste der wichtigsten Programmierhilfen, die der Echte Programmierer nicht benutzt:
- FORTRAN-Präprozessoren wie MORTAN oder RATFOR. Diese Haute Cuisine der Programmierung eignet sich hervorragend, um Müsli zu produzieren.
- Quellcode-orientierte Debugger. Echte Programmierer lesen Hexdumps.
- Compiler, die Code für Array-Indexprüfung zur Laufzeit erzeugt. Sie ersticken jede Kreativität, zerstören die meisten der interessanten Anwendungen der EQUIVALENZ-Vereinbarungen, und machen Änderungen des Betriebssystems mit Hilfe negativer Indizes unmöglich. Und schlimmer noch, solcher Code ist ineffizient.
- Programm-Pflege-Systeme. Ein echter Programmierer hält seine Software als Kartonstapel unter Verschluss, denn dies zeigt, daß der Besitzer seine wichtigsten Programme nicht unbeachtet lassen kann.

Wo der typische Echte Programmierer arbeitet? Welche Art von Programmen derart talentierter Individuen würdig ist? Nun, man kann sicher sein, daß man nie einen Echten Programmierer beim Schreiben von Buchhaltungsprogrammen in COBOL erwischen wird, oder gar beim Sortieren von Abonnementadressen des Spiegels. Nein, Echte Programmierer brauchen Aufgaben von weltbewegender Bedeutung.

Echte Programmierer arbeiten für das Los Alamos National Laboratory und schreiben dort Atomkriegs-Simulationen auf Cray-I-Supercomputern, oder sie arbeiten bei der National Security Agency und entschlüsseln russische Funksprüche. Nur weil tausende Echter Programmierer für die NASA gearbeitet haben, waren 'unsere' Jungs eher auf dem Mond als die Kosmonauten. Die Computer im Space Shuttle wurden von Echten Programmierern programmiert, und auch das Betriebssystem der Cruise Missiles der Firma BOEING wurde von diesen echten Professionals entworfen.

Einige der ehrfurchtflößendsten Echten Programmierer kennen das gesamte Betriebssystem der Pioneer- und Voyager-Sonden auswendig. Mit einer Kombination aus großen bodengebundenen FORTRAN-Programmen und kleinen, in der Sonde mitgeführten Assemblerprogrammen vollbringen sie unglaubliche Kunststücke der Navigation und Improvisation. So treffen sie nur 10 Kilometer große Fenster nahe Saturn nach 6 Jahren Flug durch den Weltraum, oder reparieren bzw. umgehen defekte Sensoren, Sender oder Batterien. Angeblich soll es einem Echten Programmierer sogar gelungen sein, in ein paar hundert Bytes ungenutzten Speichers innerhalb der Voyager-Sonde ein Mustererkennungsprogramm zu pressen, das einen neuen Mond des Jupiters suchte, fand und fotografierte.

Für die Galileo-Sonde ist vorgesehen, daß sie auf ihrem Weg zum Jupiter entlang einer schwerkraftgelenkten Bahn am Mars vorbeizieht. Diese Bahn führt in einer Entfernung von 80 ± 3 Kilometern an der Marsoberfläche vorbei. Kein Mensch würde diese Art von Navigation einem PASCAL-Programmierer oder gar -Programmierer anvertrauen.

Viele Echte Programmierer dieser Welt arbeiten für die amerikanische Regierung, meist für das Verteidigungsministerium. So soll es sein. In letzter Zeit allerdings erscheinen dunkle Wolken am Horizont der Echten Programmierer. Es scheint, als hätten einige einflußreiche Müsli-Fresser im Verteidigungsministerium entschieden, daß in Zukunft alle Verteidigungsprogramme in so einer Art von großer vereinheitlichter Programmiersprache namens ADA geschrieben werden müßten. Lange Zeit schien es, als läge ADA's Bestimmung im Verstoß gegen alle Regeln der Echten Programmierung. Es ist eine Sprache mit Strukturen, Datentypen, strenger Typenbildung und Semikola. Kurz, sie ist wie geschaffen, um die Kreativität des typischen Echten Programmierers zu verkrüppeln.

Glücklicherweise hat diese jetzt vom DoD (Department of Defence) veröffentlichte Sprache noch genügend interessante Eigenschaften, um dem Echten Programmierer eine Annäherung zu ermöglichen: sie ist unglaublich komplex, sie enthält Möglichkeiten, um mit dem Betriebssystem herumzuspielen und Speicherbereiche neu zu verteilen, und Edgar Dijkstra mag sie nicht. Dijkstra ist, wie man wissen sollte, der Autor von "GOTO's considered harmful", einem Meilenstein der Programmiermethodologie, der von PASCAL- Programmierern und Müsli-Fressern gleichermaßen bewundert wird.

- Und außerdem, ein zu allem entschlossener Echter Programmierer kann in jeder Sprache FORTRAN-Programme schreiben

Der Echte Programmierer kann allerdings auch Kompromisse in Bezug auf seine Prinzipien eingehen und an etwas geringeren Aufgaben als der Vernichtung des Lebens arbeiten, sofern er dafür entsprechend bezahlt wird. Viele Echte Programmierer schreiben z.B. Videospiele für ATARI, allerdings spielen sie nicht damit. Ein echter Programmierer weiß, wie er die Maschine jedesmal schlagen kann, und damit ist es keine Herausforderung mehr. Jeder bei Lukas-Film ist ein Echter Programmierer, denn es wäre doch verrückt, das Geld von 50 Millionen STARWARS-Fans auszuschlagen.

Der Anteil der Echten Programmierer im Bereich der Computer-Graphics ist etwas niedriger als anderswo, was wahrscheinlich daran liegt, daß noch niemand irgendeinen Nutzen der Computer-Graphics entdeckt hat. Andererseits werden Computer-Graphics überwiegend in FORTRAN abgehandelt, daher gibt es Leute, die so das Schreiben von COBOL-Programmen vermeiden.

Im allgemeinen spielt der Echte Programmierer wie er arbeitet - mit Computern. Er ist ständig darüber erheitert, daß sein Arbeitgeber ihn tatsächlich für etwas bezahlt, was er nur so zum Spaß sowieso tun würde - allerdings achtet er darauf, diese Meinung nicht zu laut zu äußern.

Gelegentlich kommt der Echte Programmierer aus dem Büro heraus, um sich ein wenig frische Luft und ein oder zwei Bierchen zu genehmigen. Hier daher einige Hinweise, wie man Echte Programmierer außerhalb des Computerraumes erkennt:

- Auf Parties stehen Echte Programmierer in der Ecke und diskutieren über Sicherheitsmaßnahmen von Betriebssystemen und wie man um sie herumprogrammiert.
- Am Strand zeichnet der Echte Programmierer Flußdiagramme in den Sand.

- Beim Fußballspielen vergleicht der Echte Programmierer die Ergebnisse mit seinem auf grünliniertem Leporello-Papier gedruckten Computer-Simulations-Ergebnissen.
- Ein Echter Programmierer geht in die Disco, um sich die Lichtorgel anzusehen.
- Bei Begräbnissen sagt der Echte Programmierer typischerweise: "Armer Hans-Helmut. Er war mit seinem Sortierprogramm schon fast fertig, als ihn der Herzinfarkt erwisch hat."
- Im Supermarkt besteht der Echte Programmierer darauf, seine Bierdosen selber über das Fenster des Barcodelesers zu schieben, weil er keinem Kassierer zutraut, dies beim ersten Versuch richtig zu machen.

In welcher Umgebung der Echte Programmierer am besten funktioniert? Nun, dies ist eine sehr wichtige Frage für die Manager von Echten Programmierern. Wenn man bedenkt, wie teuer es ist, einen von ihnen in Betrieb zu halten, dann sollte man ihn oder sie in eine optimale Arbeitsumgebung versetzen.

Der typische Echte Programmierer lebt vor dem Computerterminal. Rund um dieses Terminal liegen Ausdrucke von jedem Programm, an dem er je gearbeitet hat, sie stapeln sich grob chronologisch geordnet auf jeder ebenen Fläche des Büros. Im Zimmer verteilt finden sich über ein Dutzend mit kaltem Kaffee mehr oder weniger gefüllte Tassen. Gelegentlich schwimmen Zigarettenkippen darin herum, in einigen Fällen auch Reste von Orangenschalen. Irgendwo liegen Kopien des OS JCL-Manuals und der "Principles of Operation" herum. Über den Boden verteilt liegen Reste der Verpackung von gefüllten Keksen (der Typ, der in der Fabrik schon so furztrocken gebacken wird, daß er auch bei längerem liegen im Automaten nicht schlechter wird).

Schließlich, in der linken oberen Schublade des Schreibtisches, unter der Schachtel mit den Muntermachern, liegt eine Schablone für Flußdiagramme, die sein Vorgänger dort vergessen hat. Echte Programmierer schreiben Programme und keine Dokumentation; das überläßt man den Typen von der Wartung.

Der Echte Programmierer ist in der Lage, 30, 40, ja sogar 50 Stunden in einem Rutsch zu arbeiten und das unter hohem Zeitdruck. Genaugenommen mag er es so am liebsten. Schlechte Antwortzeiten regen den Echten Programmierer nicht auf - sie geben ihm die Chance, zwischen 2 Kommandos ein bißchen Schlaf zu ergattern. Wenn die Planung nicht genug Zeitdruck bereithält, dann tendiert der Echte Programmierer dazu, seine Arbeit herausfordernder zu machen, indem er sich die ersten neun Wochen mit einem kleinen, aber sehr interessanten Teil des Problems befaßt, um dann in der letzten Woche seine Aufgabe in nur zwei oder drei 50-Stunden-Marathonsitzungen zu beenden. Dies beeindruckt nicht nur den Manager, sondern schafft gleichzeitig eine hervorragende Entschuldigung für das Fehlen der Dokumentation.

Und überhaupt: Kein Echter Programmierer arbeitet von 9 bis 5, außer denen von der Nachtschicht. Echte Programmierer tragen keine Schlipse. Echte Programmierer tragen keine hochhackigen Schuhe. Echte Programmierer kommen zur Arbeit, wenn andere zum Mittagessen gehen. Ein echter Programmierer vergißt vielleicht den Vornamen seiner Angetrauten, aber niemals den Inhalt der gesamten ASCII- oder EBCDIC-Tabelle. Echte Programmierer können nicht kochen. Da Supermärkte um 3 Uhr morgens selten geöffnet sind, müssen sie sowieso von Kaffee und Keksen leben.

Die Zukunft betrachtend machen sich eine ganze Reihe von Echten Programmierern Sorgen, daß die jüngste Programmierergeneration nicht mehr mit der gleichen Lebensperspektive aufwächst, wie sie selbst. Viele der Jüngeren haben noch nie einen Computer mit Maschinenkonsole gesehen. Kaum ein Schulabgänger kann heute noch hexadezimal rechnen, ohne einen Taschenrechner zu benutzen. Die Programmierung durch symbolische Debugger oder Text-Editoren, die Klammern zählen, und benutzerfreundliche Betriebssysteme. Und das Schlimmste ist, einige von ihnen werden auf die Menschheit losgelassen, ohne je FORTRAN zu lernen! Sind wir dazu verdammt, eine Industrie von UNIX-Hackern und PASCAL-Programmierern zu werden?

Nun, aus meiner Erfahrung glaube ich behaupten zu dürfen, daß das Schicksal den Echten Programmierern Wohlgesonnen ist. Weder OS/370 noch FORTRAN zeigen irgendwelche Symptome des Aussterbens, trotz aller Anstrengung der PASCAL-Programmierer. Selbst subtile Tricks, wie das Hinzufügen strukturierter Schleifen zu FORTRAN sind fehlgeschlagen. Sichter, einige Hersteller liefern FORTRAN-77- Compiler, aber jeder einzelne von ihnen läßt sich über eine einzige Compileroption in einen FORTRAN-66-Compiler verwandeln - mit Schleifen, wie Gott sie schuf.

Selbst UNIX scheint für den Echten Programmierer nicht mehr so schlecht zu sein wie früher. Die neueste UNIX-Version hat das Potential eines Betriebssystems, das eines Echten Programmierers würdig ist. Sie hat zwei verschiedene, leicht inkompatible Benutzerschnittstellen, einen geheimnisvollen und komplizierten Teletype-Treiber und virtuellen Speicher. Und wenn der Echte Programmierer die Strukturierung ignoriert, kann er sich sogar mit C anfreunden. Schließlich gibt es keine Typenbindung, Bezeichner sind sieben (acht? zehn?) Zeichen lang und man hat Zeiger als Bonus. Das ist, als hätte man die besten Teile von FORTRAN und Assembler vereint, von den kreativen Möglichkeiten des #define ganz zu schweigen.

Nein, die Zukunft ist nicht völlig schlecht. So hat sich in den vergangenen Jahren die populäre Presse sogar über neue Brut von cleveren Computer-Schratzen und -Hackern geäußert, die Plätze wie Stanford oder das MIT zugunsten der Wirklichkeit verlassen haben. Allen Anzeichen nach lebt der Geist der Echten Programmierer weiter in diesen jungen Männern und Frauen. Und solange es schlecht beschriebene Ziele, bizarre Fehler und unrealistische Zeitpläne gibt, solange wird es Echte Programmierer geben, die bereit sind einzuspringen und das Problem zu lösen, und die sich die Dokumentation für später aufheben.

(Aus DATAMON 7/83. Übersetzt von Bernd Leeds. Eingespeist ins S&F-Netz von Bernd, DHSIAE @ DB0GE-8. Erstmals eingespielt 1989 von Arno, DG2EAT)