

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 Programmierer sagte Dinge wie "DO 10 I=1, 10" oder "ABEND", und der Rest der Welt quengelte "Computer sind mir zu kompliziert" oder "Ich kann zu Computern keine gefühlsmäßige aufbauen - sie sind so unpersönlich". Dabei zeigte schon Remy Eyssen's 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 in einer Welt gegenüber, in der kleine alte Damen vollcomputerisierte Mikrowellenherde kaufen können, in der zwölf Jahre alte Dreikäsehochs gestandene Männer bei ASTEROIDS und PACMAN sattmachen, und in der jeder seinen eigenen Heimcomputer kaufen und sogar verstehen kann. Der Echte Programmierer ist gefährdet von Studenten mit einem igITT 2020 (deutsche Version des ABFALL-II, Anmerkung des Übersetzers) im Gepäck ersetzt zu werden!

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

Der einfachste Weg um einen Echten Programmierer zu erkennen führt über die von ihm verwendete Programmiersprache. Echte Programmierer benutzen FORTRAN. Müslifresser benutzen Pascal. Nikolaus 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 Parameterübergabemechanismus, den Echte Programmierer akzeptieren ist call-by-value-return (call-by-result, Anmerkung des Übersetzers), wie er in den IBM 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 Lochkartenstanzer, einem FORTRAN IV Compiler und einem Bier. Echte Programmierer erledigen Listenverarbeitung, Zeichenkettenmanipulation, Abrechnungswesen (wenn überhaupt) und künstliche Intelligenz in FORTRAN. Was sie nicht in FORTRAN machen können machen sie in Assembler, was sie nicht in Assembler machen können lassen sie verächtlich liegen.

Akademische Computerspezialisten sind in den letzten Jahren auf das Abstellgleis der strukturierten Programmierung geraten. Sie behaupten, daß Programme verständlicher werden, wenn bestimmte Sprachkonstrukte und Programmieretechniken benutzt werden. Sie können sich natürlich nicht er-

nigen, 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, Anmerkund des Übersetzers) schreiben, beherrschte fünf 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 zwei zu beschleunigen. Jeder Echte Programmierer wird einem versichern, daß die gesamte strukturierte Programmierung in diesem Fall nicht hilft - hier braucht man wirklich Talent.

Einige Betrachtungen zum Thema "Echte Programmierer und strukturierte Programmierung":

- Echte Programmierer haben keine Angst vor GOTO's.
- Echte Programmierer schreiben fünf Seiten lange DO-Schleifen, ohne durcheinander zu geraten.
- Echte Programmierer lieben das arithmetische IF-Statement (das mit den drei Ausgängen, Anmerkung des Übersetzers), weil sie den Code interessanter machen.
- Echte Programmierer schreiben selbstmodifizierende Programme, speziell wenn sie damit in einer kleinen Schleife zwanzig Nanosekunden einsparen können.
- Echte Programmierer brauchen keine Kommentare, das Programm ist selbstdokumentierend.
- Da FORTRAN strukturierte IF, REPEAT...UNTIL oder CASE Anweisungen nicht kennt, braucht sich der Echte Programmierer nicht zu sorgen, daß er sie nicht benutzt. Außerdem kann man sie nötigenfalls über "Assigned GOTO's" simulieren.

Auch Datenstrukturen waren in letzter Zeit in der Diskussion. Abstrakte Datentypen, Records, Pointer, Listen und Zeichenketten sind in gewissen Kreisen populär geworden. Wirth, der Müslifresser, verfasste sogar ein ganzes Buch ("Algorithmen und Datenstrukturen", Teubner 1975), in dem er behauptete, daß man Programme schreiben könne, die auf Datenstrukturen aufbauen, statt es umgekehrt zu machen. Wie jeder Echte Programmierer weiß gibt es nur eine wirklich sinnvolle 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 Typen ist außerdem, daß man sie deklarieren muß, während Echte Programmiersprachen, wie man weiß, den Typ anhand des ersten Buchstaben eines maximal sechs Zeichen langen Bezeichners implizit festlegen.

Welches Betriebssystem der Echte Programmierer benutzt? CP/M? Gott bewahre! Das ist doch im Grunde ein Spielzeug-Betriebssystem. Selbst kleinere 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 diese Woche heißt - aber wenn man es genau nimmt, ist UNIX doch nur ein verherlichtes Telespiel. Niemand arbeitet auf UNIX Systemen an ernstzunehmenden Dingen - man schickt kleine Witzchen über USENET rund um die Welt, oder schreibt ein neues Adventure Spiel oder Forschungsberichte.

Nein der Echte Programmierer benutzt OS/370. Ein guter Programmierer kann die Beschreibung des Fehlers IJK3051 in seinem JCL Handbuch finden und verstehen. Ein großartiger Programmierer kann JCL schreiben, ohne je in das Handbuch zu sehen. Ein wahrhaft außergewöhnlich guter Programmierer kann Fehler in einem sechs Megabyte Hexdump finden, ohne einen Taschenrechner zu benutzen.

OS/370 ist wirklich ein bemerkenswertes Betriebssystem. Mit einem einzigen falsch platzierten Leerzeichen kann man die gesamte 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 ein Echter 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 Systemurlader 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 sollten oder halten Informationen, die schon lange weg sein sollen. Aber zurück zum Thema. Die Legende sagt, daß Seymour Cray, der Erfinder des Cray-1 Supercomputers und der meisten Computer von Control Data, selbst das erste Betriebssystem für die CDC7600 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, arbeitete 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 das Telephon. Er brachte den Benutzer dazu, über die Maschinenkonsole Disk I/O Instruktionen einzutasten, Systemtabellen in Hexadezimal zu reparieren und Registerinhalte über das Telephon durchzugeben. Die Moral von der Geschichte: Obwohl ein Echter Programmierer normalerweise Kartenlocher und Schnelldrucker benutzt, kommt er im Notfall auch mit Maschinenkonsole und Telephon aus.

In einigen Firmen besteht die Programmeingabe allerdings nicht mehr aus zehn schlangestehenden Ingenieuren, die auf einen 029-Locher warten. In

meiner Firma zum Beispiel steht kein einziger Kartenlocher. Der Echte Programmierer muß in diesem Falle seine Arbeit mit einem Texteditor 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 wurden und auf Alto oder Dorado Computern laufen. Unglücklicherweise würde jedoch kein Echter Programmierer einen Computer mit einem Betriebssystem benutzen, daß SmallTalk heißt und sicher 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 IV. Das Problem mit diesen Editoren ist, daß Echte Programmierer das Konzept des "Du kriegst, was du siehst" für schlecht halten. Der Echte Programmierer will einen "Du hast es soll gewollt, da hast Du's"-Editor, einen der kompliziert ist, kryptisch, 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 dank TECO 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 Objektcode direkt zu ändern, für gewöhnlich mit einem 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-Quellprogramm haben. In einigen Fällen ist nicht einmal mehr das 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 Müsli-fressender strukturierter Programmierer wüßte auch nur, wo er mit der Arbeit anfangen soll. Man nennt das Arbeitssicherungsmaßnahmen.

Hier eine Liste der wichtigsten Programmierhilfen, die der Echte Programmierer nicht benutzt:

- FORTRAN-Präprozessoren wie MORTRAN oder RATFOR. Dies Haute Cuisine der Programmierung eignet sich hervorragend, um Müsli zu produzieren.
- Quellcodeorientierte Debugger. Echte Programmierer lesen Hexdumps.
- Compiler, die Code für Array-Indexüberprüfungen erzeugen. Sie erstickten jede Kreativität im Keim; zerstören die meisten der interessantesten Anwendungen der EQUIVALENCE-Vereinbarungen, und machen

Änderungen des Betriebssystems mit Hilfe negativer Indizes unmöglich. Und schlimmer noch, solcher Code ist ineffizient.

- Programmpflegesysteme. Ein Echter Programmierer hält seine Software als Kartenstapel unter Verschluss, denn dies zeigt, daß der Besitzer seine wichtigen Programme nicht unbewacht lassen kann.

Wo der typische Programmierer arbeitet? Welche Art von Programmen derart talentierter Individuen würdig sind? Nun, man kann sicher sein, daß man nie einen Echten Programmierer beim schreiben von Buchhaltungsprogrammen in COBOL erwischen wird, oder gar beim sortieren von Abonnentenadressen des STERN-Magazins. Nein ein Echter Programmierer braucht Aufgaben von weltbewegender Bedeutung. Echte Programmierer arbeiten für das Los Alamos National Laboratory und schreiben dort Atomkriegs-Simulationen auf Cray-1-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 die Betriebssysteme der Cruise Missiles der Firma BOEING wurden von diesen echten Professionals entworfen.

Einige der furchteinflößendsten Echten Programmierer arbeiten im Jet Propulsion Laboratory in Kalifornien. Viele von ihnen kennen das gesamte Betriebssystem der Pioneer- und Voyagersonden auswendig. Mit einer Kombination aus großen, bodengebundenen FORTRAN-Programmen und kleinen, von den Sonden mitgeführten Assemblerprogrammen vollbringen sie unglaubliche Kunststücke der Navigation und Improvisation. So treffen sie nur 10 Kilometer große Fenster nahe Saturn nach sechs Jahren Flug durch den Weltraum, oder reparieren beziehungsweise umgehen defekte Sensoren, Sender oder Batterien. Angeblich soll es einem Echten Programmierer sogar gelungen sein, in ein paar hundert Bytes unbenutzten Speichers innerhalb der Voyager-Sonde ein Mustererkennungsprogramm zu pressen, daß einen neuen Mond des Jupiters suchte, fand und photographierte. 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 km an der Marsoberfläche vorbei. Kein Mensch würde diese Art von Navigation einem Pascal-Programm oder gar -Programmierer anvertrauen.

Viele der Echten 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 Müslifresser 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 Typenbindung und Semikolons. Kurz, sie ist wie geschaffen um die Kreativität des typischen Echten Programmierers zu verkrüppeln.

Glücklicherweise hat die jetzt vom DoD ausgewählte Sprache noch genügend interessante Eigenschaften, um dem echten Programmierer eine Annäherung zu ermöglichen: Sie ist unglaublich komplex, sie enthält Möglichkeiten um mit dem Betriebssystem herumzumachen 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 Müslifressern 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 geringerem als der Vernichtung des Lebens arbeiten, sofern er dafür entsprechend bezahlt wird. Viele echte Programmierer schreiben zum Beispiel 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 Lucas-Film ist ein Echter Programmierer, denn es wäre doch verrückt, das Geld von fünfzig Millionen Star-Wars Fans auszuschlagen.

Der Anteil der Echten Programmierer im Bereich der Computer-Graphics ist etwas niedriger als anderswo, was wahrscheinlich daran liegt, daß noch niemand den Nutzen der Computer-Graphics entdeckt hat. Andererseits werden Computer-Graphics überwiegend in FORTRAN abgehandelt, daher gibt es einige 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 auch aus seinem Büro heraus, um sich ein wenig frische Luft und ein oder zwei Bierchen zu genehmigen. Hier einige Hinweise, wie man den Echten Programmierer außerhalb des Computerraums erkennt:

- Auf Parties stehen Echte Programmierer in einer Ecke und diskutieren über die Sicherheitsmechanismen von Betriebssystemen, und wie man darum herumprogrammiert.
- Bei Fußballspielen vergleicht der Echte Programmierer die Ergebnisse mit seinen auf grünliniertem Leporello-Papier gedruckten Computer Simulationsergebnissen.
- Am Strand zeichnet ein Echter Programmierer Flußdiagramme in den Sand.
- 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 erwischt hat."
- Im Supermarkt besteht der Echte Programmierer darauf, seine Bierdosen selber über das Fenster des Strichcodelesers 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 im Betrieb zu halten, dann sollte man ihn oder sie in eine optimale Arbeitsumgebung versetzen.

Der typische Programmierer lebt vor einem Computerterminal. Rund um dieses Terminal liegen Ausdrücke 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 Zigarettenskippen darin herum, in einigen Fällen auch Reste von Orangenschalen. Irgendwo liegen Kopien des OS JCL Manuals und der "Principles Of Operation" an einer besonders interessanten Stelle aufgeschlagen herum, außer bei extrem guten Leuten. An der Wand klebt ein Schnelldrucker-Kalender mit Snoopy drauf aus dem Jahr 1969. Über den Boden verteilt liegen Reste der Verpackungen von gefüllten Keksen (der Typ, der schon in der Fabrik so furztrocken gebacken wird, daß er auch bei längerem Liegen im Automaten nicht schlechter wird). Schließlich, in der linken, oberen Schublade des Schreibtischs, unter der Schachtel mit den Muntermachern, liegt eine Schablone für Flußdiagramme, die sein Vorgänger dort vergessen hat. Echte Programmierer schreiben Programme und keine Dokumentation, daß überläßt man den Typen von der Wartung.

Der Echte Programmierer ist in der Lage dreißig, vierzig, ja sogar fünfzig 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 zwei 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 zwei oder drei fünfzig Stunden Marathonsitzungen zu beenden. Dies beeindruckt nicht nur den Manager, sondern schafft gleichzeitig eine hervorragende Entschuldigung für das Fehler der Dokumentation. Und überhaupt kein Echter Programmierer arbeitet von neun bis fünf 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 Namen seiner Angetrauten, aber niemals den Inhalt der gesamten ASCII- (oder EBCDIC-) Tabelle. Echte Programmierer können nicht kochen. Da Super-

märkte um drei Uhr morgens selten geöffnet sind, müssen sie sowiso von Kaffee und Keksen leben.

Die Zukunft betrachtend machen sich eine 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 einer Maschinenkonsole gesehen. Kaum ein Schulabgänger kann heute noch hexadezimal rechnen, ohne einen Taschenrechner zu benutzen. Die Studenten von heute sind weich - geschützt von den Realitäten der Programmierung durch symbolische Debugger, Texteditoren die Klammern zählen und benutzerfreundliche Betriebssysteme. Und das Schlimmste ist, einige dieser angeblichen Computer-Spezialisten kommen zu Rang und Namen ohne je FORTRAN zu lernen! Sind wir dazu verdammt, eine Industrie von UNIX-Hackern und Pascal-Programmierern zu werden?

Nun, aus meiner Erfahrung heraus 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 Anstrengungen der Pascal-Programmierer. Selbst subtilere Tricks wie das Hinzufügen strukturierter Schleifen zu FORTRAN sind fehlgeschlagen. Sicher einige Computerhersteller liefern FORTRAN-77 Compiler, aber jeder einzelne von ihnen läßt sich über eine einzige Compiler-Option in einen FORTRAN-66 Compiler verwandeln - mit DO-Schleifen, wie von Gott geschaffen.

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 (Zehn? Acht?) 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 die clevere neue Brut von Computerschrauten und Hackern geäußert, die Plätze wie Stanford oder das MIT zugunsten der Wirklichkeit verlassen haben. Allen Anzeichen nach lebt der Geist der Echten Programmierung weiter in diesen jungen Männer und Frauen. Und solange es schlecht beschriebene Ziele, bizarre Fehler und unrealistische Zeitpläne gibt, solange wird es Echte Programmierer geben, die bereit sind einzuspringen, das Problem zu lösen und sich die Dokumentation für später aufheben. Lang lebe FORTRAN!