

Immer wieder war jur-pc Forum für die Diskussion um den Urheberrechtsschutz von Computerprogrammen. Erst in jüngster Zeit lebte die Debatte – ausgehend von der Nixdorf-Entscheidung des BGH (I ZR 139/89, jur-pc 01/1991, S. 888) – mit dem Editorial von Herberger, („Das Lesen eines Buches, das Anhören einer Schallplatte, das Betrachten eines Kunstwerks ...“, jur-pc 01/1991, S. 887), der in der BGH-Entscheidung einen neuen Kurs im Software-Recht ausmachen konnte, sowie der Anmerkung Walts (jur-pc 04/1991, S. 1059) wieder auf. König setzt sich mit der zentralen These – Computerprogramme als Sprachwerke – von Clapes/Lynch/Steinberg (Das Epos von Silicon und die Barden des Binären jur-pc 8/1989, S. 279ff) engagiert auseinander. Entgegen der bisher in jur-pc überwiegend vertretenen Auffassung kommt er zu dem Ergebnis, daß Computerprogramme keine Sprachwerke sein können.

## Computerprogramme sind keine Sprachwerke Teil 1

Michael König

### Einleitung

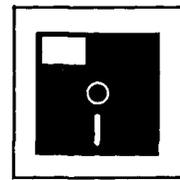
Die Diskussion um den urheberrechtlichen Schutz von Computerprogrammen ist noch nicht beendet. Daran haben weder die Urheberrechtsnovelle von 1985<sup>1</sup> noch die „Inkasoprogramm“-Entscheidung des BGH<sup>2</sup> etwas zu ändern vermocht. Eine für den Urheberrechtsschutz wichtige Frage ist, ob Computerprogramme Sprachwerke i.S.d. § 2 I Nr. 1 sind. Hierzu ist in jur-pc ein streitbarer Artikel von Clapes, Lynch und Steinberg erschienen<sup>3</sup>, in dem diese die These vertreten, daß Computerprogramme literarische Sprachwerke seien. Sowohl dieser These als auch der zu deren Begründung vorgetragenen Argumentation muß entschieden entgegengetreten werden. Hierbei bleibt allerdings die 1985 erfolgte Aufnahme von Computerprogrammen in § 1 I Ziff. 1 UrhG zunächst unberücksichtigt; deren Auswirkungen werden erst abschließend erörtert<sup>4</sup>.

I.

Zunächst sind einige Vorbemerkungen erforderlich. Um kein Mißverständnis entstehen zu lassen muß ausdrücklich klargestellt werden, daß auch der Verfasser die Schutzbedürftigkeit von Computerprogrammen selbstverständlich unbedingt bejaht. Auch der fanatischste Raubkopierer kann sich nichts mehr als einen möglichst weitgehenden Schutz von Computerprogrammen vor Raubkopieren wünschen, wenn er auch nur etwas weiter als bis zu seinem Monitor bzw. seiner Diskettenbox denkt. Ohne effektiven Schutz und damit die Chance der Amortisation der erfolgten Investitionen wird sich kaum noch jemand zur Herstellung von des Raubkopierens werten Computerprogrammen bereithalten. Der Schutz darf indes nur soweit reichen, wie er wie er durch die Interessen der Programmhersteller gerechtfertigt ist. Keinesfalls ist es daher legitim, die Unterstellung von Computerprogrammen unter ein bestimmtes Schutzinstrument mit dem Ziel zu fordern, hierdurch eine möglichst umfassende Kontrolle über den gesamten EDV-Bereich auszuüben.

Hieran anknüpfend darf nicht übersehen werden, daß die Diskussion, ob Computerprogramme Sprachwerke oder... (was ?) darstellen, überwiegend nicht der Fortbildung des Rechts oder gar der intellektuellen Befriedigung wissenschaftlich interessierter Geister dient. Hintergrund sind vielmehr handfeste wirtschaftliche Interessen, die über das legitime Ziel, das Kopieren von Programmen und damit den Verlust der Amortisation erfolgter Investitionen zu verhindern, weit hinausgehen.

Beabsichtigt ist zunächst, den Schutz des Inhalts von Computerprogrammen zu gewährleisten – eines Schutzes, für den das deutsche Urheberrecht bei technischen Produkten wie Computerprogrammen nicht auch nur entfernt bestimmt und geeignet ist. Zum anderen soll gerade auch die Benutzung von Computerprogrammen urheberrechtlich erfaßt und entsprechend kontrolliert werden. Unerheblich ist, ob man dies über die



*Der Stand der Diskussion*

*Unstreitig: Schutzbedürftigkeit von Computerprogrammen*

*Hintergrund: Wirtschaftliche Interessen*

*Angestrebte Schutzbereiche ...*

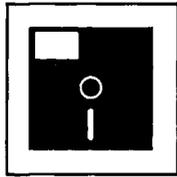
<sup>1</sup> BT-Drucksache 10/3360, 18

<sup>2</sup> BGH I ZR 52/83 v. 9.5.85 NJW 86, 192

<sup>3</sup> Clapes/Lynch/Steinberg, Das Epos von Silicon und die Barden des Binären, jur-pc 89, 219ff, 230ff, 279ff

<sup>4</sup> hierzu unten bei V.

Michael König ist Rechtsanwalt in Frankfurt.



... und die Folgen

Argumentation zu erreichen versucht, der Schutz dürfe nicht dort aufhören, wo die wichtigste Verwertungshandlung beginne (weil nicht sein kann was nicht sein darf)<sup>5</sup>, oder ob man technisch bedingte Vorgänge wie die wegen der von-Neumann-Architektur der Rechner (noch) regelmäßig erforderliche Herstellung eines Duplikats im Hauptspeicher zum Anlaß nimmt, einer urheberrechtlich relevanten Vervielfältigung das Wort zu reden<sup>6</sup>. In jedem Fall würde als Ergebnis jede einzelne Programm Benutzung kontrolliert werden können. Dies wäre nicht nur für den bei der Herstellung nicht beteiligten Eigentümer einer Raubkopie von Bedeutung, der dadurch entgegen der eindeutigen gesetzgeberischen Wertung aus § 101 UrhG die Raubkopie nicht mehr benutzen könnte, sondern hätte auch ganz erhebliche Auswirkungen auf den Markt für „gebrauchte“ bzw. alte Computerprogramme und gebrauchte Rechner: Wer ein „altes“ Computerprogramm, also z.B. eine weniger leistungsfähige Vorversion, „gebraucht“ erwirbt, bedürfte zu jeder Programm Benutzung die Genehmigung des Herstellers – die dieser selbstverständlich nur gegen ein weiteres Entgelt erteilen würde<sup>7</sup>. Damit nicht genug: Rechner brauchen ein Betriebssystem. Dieses wird in vielen Rechnerklassen nicht separat angeboten, sondern nur zusammen mit der Hardware geliefert<sup>8</sup>. Wenn nun die Benutzung des zusammen mit dem Rechner gebraucht erworbenen Betriebssystems untersagt werden könnte, andererseits – wie wohl noch in den meisten Marktsegmenten der Fall – kein „freies“ Betriebssystem angeboten wird, so kann der Hersteller bzw. Lieferant von Rechner und Betriebssystem den Handel mit gebrauchten Geräten seiner Herstellung und damit die Konkurrenz mit seinen Neugeräten verhindern und den Absatz der letztgenannten fördern<sup>9</sup>. Das hiermit besonders angesprochene Unternehmen braucht namentlich nicht genannt zu werden.<sup>10</sup>

## II.

*Die Argumentation von Clapes, Lynch und Steinberg*

Clapes, Lynch und Steinberg geben sich große Mühe, den Charakter von Computerprogrammen als „literarische Sprachwerke“ zu belegen. Reduziert man jedoch deren Beitrag auf den Kern, so bleibt letztlich als Argumentation übrig, daß Computerprogramme Sprachwerke seien, weil sie von Autoren geschrieben würden, weil der Herstellungsprozeß sowie die Lage der Autoren ähnlich sei wie bei der Herstellung von Büchern und – inzident – weil Computerprogramme eben (und selbstverständlich) aus Sprache bestünden. Auffällig ist, daß Clapes, Lynch und Steinberg hierbei dieselbe „strategische Wortverwendung“ praktizieren, wie sie sie den Vertretern der Gegenauffassung vorwerfen<sup>11</sup>. Durch die ständige Verwendung der Begriffe „Sprache“, „schreiben“, „Autor“, „Ausdruck“ und „Werk“ suggerieren sie das an sich erst zu begründende Ergebnis als richtig und feststehend.

*Die „Lesbarkeit“ von Listings ...*

Das eigentliche Argument ist, daß Computerprogramme, und zwar Quellcodes, „gelesen“ werden könnten. Dies ist nicht auf den Beitrag von Clapes, Lynch und Steinberg beschränkt; das Argument der „Lesbarkeit“ ist z.B. auch offensichtlich die Leitlinie des Editoriais Herbergers aus jur-pc 3/90<sup>12</sup>.

<sup>5</sup> Kollé, Bericht der deutschen Landesgruppe für die Tagung des Geschäftsführenden Ausschusses vom 13. Mai 1985 in Rio de Janeiro, GRUR Int. 85, 29, 31; ders., Der Rechtsschutz der Computersoftware in der Bundesrepublik Deutschland, GRUR 82, 443, 445; ähnlich auch Kindermann, Vertrieb und Nutzung von Computer-Software aus urheberrechtlicher Sicht, GRUR 83, 150, 158

<sup>6</sup> z.B. Kindermann a.a.O. S.156f; Ulmer/Kollé, Der Urheberrechtsschutz von Computerprogrammen, GRUR Int. 82, 489, 498; dagegen z.B. Betten, Urheberrechtsschutz von Computerprogrammen?, Mitt. 84, 201, 203f; differenzierend Haberstumpf, Der Ablauf eines Computerprogramms im System der urheberrechtlichen Verwertungsrechte, CR 87, 409.411f; König, Das Computerprogramm (erscheint 1991), C.II.3.c)aa); Loewenheim in Schrickler Rdnr. 9 zu § 16

<sup>7</sup> Engel, Der Software-Verletzungsprozeß, in: Lehmann, Rechtsschutz und Verwertung von Computerprogrammen, 573, Rdnr. 11; Wolf, Rechtsprobleme beim Konkurs des Softwareherstellers, in: Kilian/Gorny, Schutz von Computer-Software, 81, 88 ff., 95 Anmerkung 13; Zahmt, Einsatz von Standardanwendungsprogrammen auf „fremden“ DV-Anlagen, CR 89, 965, 967

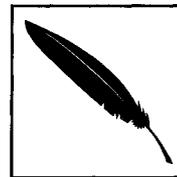
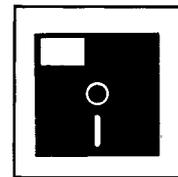
<sup>8</sup> Bis vor kurzem war dies i.ü. auch bei MS-DOS-bzw. PC-DOS-PC der Fall: Weder MS-DOS noch PC-DOS konnten offiziell separat erworben werden. Erst die Einführung von DRDOS schaffte Abhilfe.

<sup>9</sup> vgl. hierzu OLG München 29 U 2036/87 v. 14.1.88 RDV. 88, 87; OLG Nürnberg 3 U 1342/88 v. 20.6.88 CR 90, 118, 121; LG Bielefeld 20 O 412/84 v. 18.4.86 IuR 86, 459 mit Anmerkungen von Zahmt sowie Kindermann CR 86, 446; Köhler H., Rechtsfragen zum Softwarevertrag, CR 87, 827ff, 833; Lehmann, Aktuelle kartell- und wettbewerbsrechtliche Probleme der Lizenzierung von urheberrechtlich geschützten Computerprogrammen, BB 85, 1209f; Schneider J., Softwarenutzungsverträge im Spannungsfeld von Urheber- und Kartellrecht, 85f

<sup>10</sup> hierzu auch Zahmt, Willkommen IBM!, IuR 86, 432f

<sup>11</sup> Clapes/Lynch/Steinberg jur-pc 89, 218, 221 Fußnote 19

<sup>12</sup> Herberger, Listings lesen, jur-pc 90, 483



Hierbei wird indes übersehen, daß z.B. auch elektronische Schaltpläne „gelesen“ werden können – freilich nur von demjenigen, der die „Sprache versteht“, also um die Bedeutung der verschiedenen Symbole weiß und über das erforderliche elektronische Wissen verfügt. Es ist jedoch offensichtlich, daß elektronische Schaltpläne weder Sprachwerke noch Literatur sind; sie sind (selbstverständlich) unbestritten Darstellungen technischer Art gem. § 2 I Nr.7 UrhG und als solche möglicherweise schützbar – nämlich nur dann, wenn die jeweilige Darstellungsform den an eine persönliche geistige Schöpfung zu stellenden Anforderungen des § 2 II UrhG genügt. Es darf hier als bekannt vorausgesetzt werden, daß der Inhalt von Darstellungen technischer Art weder zur Begründung des Urheberrechtsschutzes herangezogen werden darf noch von diesem umfaßt wird<sup>13</sup>; die konkrete elektronische Schaltung, also das Schaltungsprinzip und die eigentliche Leistung des Schaltungsentwicklers, ist somit nicht durch das Urheberrechtsgesetz geschützt.

Auch aus urheberrechtlicher Sicht unterscheidet man daher zwischen der elektronischen Schaltung als Schaltplan und der konkreten Zusammenschaltung unterschiedlicher elektronischer Bauteile<sup>14</sup>. Diese Zusammenschaltung der Bauteile – obwohl inhaltlich mit dem Schaltplan identisch – ist selbstverständlich weder eine Darstellung technischer Art noch gar ein Sprachwerk – sie kann noch nicht einmal als ein Werk i.S.d. Urheberrechts betrachtet werden, denn sie ist nur die unbedingt freie Ausführung und Realisierung des Schaltplans. Elektronische Schaltungen unterscheiden sich aus urheberrechtlicher Sicht in keiner Weise von sonstigen Erzeugnissen, die nach Vorlage einer Konstruktionszeichnung oder eines Bauplans hergestellt werden<sup>15</sup>.

Demgegenüber fehlt eine entsprechende sachgerechte Unterscheidung, wenn von Computerprogrammen als Sprachwerken oder gar „literarischen Werken“ gesprochen wird. Die Differenzierung zwischen Quellcodes und Maschinenprogrammen – auch als Objektcode oder -programm bezeichnet – wird regelmäßig in ihrer Bedeutung nicht erkannt; die behauptete Sprachwerkeigenschaft der Quellcodes wird pauschal auf Maschinenprogramme übertragen, allenfalls kaschiert mit dem Anschein einer Begründung, wonach das Maschinenprogramm aus einer Reihe von Einsen und Nullen<sup>16</sup> oder hexadezimalen Zahlen<sup>17</sup>, also im Ergebnis auch Sprachzeichen, bestünde.

Diese Betrachtungsweise ist unzutreffend. Richtigerweise können Quellcodes allenfalls als Darstellungen technischer Art betrachtet werden; Maschinenprogramme hingegen zählen ebensowenig wie elektronische Schaltungen als Zusammenschaltung der einzelnen Bauteile, Rasierapparate oder Atomkraftwerke zu den urheberrechtlich schützba- ren Werken. Zur näheren Begründung ist ein Eingehen auf computertechnische Grundlagen erforderlich, das einige (bewußte?) Unklarheiten und Unrichtigkeiten der Darstellung von Clapes, Lynch und Steinberg korrigiert.

### III.

#### 1.

Maschinenprogramme bestehen mitnichten aus Einsen und Nullen oder sonstigen Zeichen. Maschinenprogramme sind vielmehr eine bestimmte Zusammenschaltung von

... und von elektronischen  
Schaltplänen

*Quellcode und Objektcode*

*Maschinenprogramme:  
Miniaturisierte Äquivalente elek-  
tronischer Bauteile*

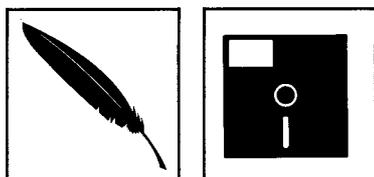
<sup>13</sup> einhellige Auffassung, vgl. nur BT-Drucksache IV/270, 37; BGH I ZR 52/83 v. 9.5.85 NJW 86, 192, 196; BGH I ZR 26/77 v. 15.12.78 GRUR 79, 464, 465; BGH I ZR 85/82 v. 10.5.84 GRUR 85, 129, 131; Betten Mitt. 84, 201, 206; ders., Zum Patentschutz von Computerprogrammen, Mitt. 83, 62, 67f; Dietz, Die kategoriale Herausforderung des Urheberrechts, BIE 83, 305, 307, 310; Hubmann § 15 II; Ilzhöfer, Reverse-Engineering von Software und Urheberrecht, CR 90, 578, 581; Loewenheim in Schricker Rdnr.124 zu § 2; Moser, Der urheberrechtliche Schutz von wissenschaftlich-technischen Darstellungen in Deutschland und Großbritannien, 251ff, 260; Reimer GRUR 80, 572, 578; anders jedoch im britischen Urheberrecht, welches auch nach technischen Zeichnungen hergestellte Maschinenteile u.ä. schützt: hierzu Betten Mitt. 84, 201, 206; Dietz a.a.O. S. 310; ausführlich Moser 147f, 181f, 192, 295ff, 301.

<sup>14</sup> Zur dreifachen Bedeutung des Begriffs „Schaltung“ – das logische Schaltungsprinzip – das Schaltbild – der konkrete Zusammenbau – s. v. Falckenstein, Der Schutz von Computerprogrammen nach dem Gebrauchsmusterrecht und dem Geschmacksmusterrecht, in: Lehmann, Rechtsschutz und Verwertung von Computerprogrammen, 172, Rdnr.9.

<sup>15</sup> Eine Ausnahme gilt lediglich bei Bauwerken, da hier auch das Bauwerk selbst zu den Werken i.S.d. § 2 I Nr. 4 UrhG zählt: BGH I ZR 85/82 v. 10.5.84 GRUR 85, 129, 131; Moser 267f.

<sup>16</sup> Clapes/Lynch/Steinberg jur-pc 89, 279, 280; Kullmann, Der Schutz von Computerprogrammen und -chips in der BRD und den USA, 28, 93; Zahmt, DV-Verträge: Rechtsfragen und Rechtsprechung, 29; ähnlich Maier, Zur Kaufmannseigenschaft von Software-Entwicklern, NJW 86, 1909, 1911 Fußnote 25

<sup>17</sup> Engel, Mängelansprüche bei Software-Verträgen, BB 85, 1159, 1160; Hoeren, Softwareüberlassung als Sachkauf, Rdnr. 117; Kindermann GRUR 83, 150; ders., Was ist Computer-Software?, ZUM 85, 2, 3; Preuß, Der Rechtsschutz von Computerprogrammen nach dem Urteil des BGH vom 9. Mai 1985, 10.



*Die Erscheinungsformen von  
Maschinenprogrammen*

*Maschinenprogramme der „EDV-  
Frühzeit“*

*Maschinenprogramm:  
Konfigurierter Steuerungsteil eines  
Computers*

*Quellcode: Grafische Darstellung  
für Maschinenprogramme*

elektronischen Bauteilen bzw. deren miniaturisiertes Äquivalent, sei es in statischer (ROM<sup>18</sup>) oder in dynamischer (RAM<sup>19</sup>) Form<sup>20</sup>. Sie können auch als ein auf eine bestimmte Art magnetisiertes Stück Magnetplatte, -scheibe oder -band (Diskette, Festplatte oder Magnetband) in Erscheinung treten. Es gibt sie ferner in mechanisch-„(be)greifbarer“ Ausführung, nämlich z.B. als CD-ROM, bei der bekanntlich die Programme durch das Vorhandensein oder Nichtvorhandensein von Löchern im Trägermedium determiniert werden – hier ist der Vergleich mit Nocken-Schaltwerken<sup>21</sup> besonders plastisch.

Da infolge der Digitaltechnik alle computerinternen Vorgänge mit „Spannung-an“ und „Spannung-aus“, symbolisch „1“ und „0“ oder „H“ (für „High“ = 5 Volt) und „L“ (für „Low“ = 0 Volt) beschrieben werden, lassen sich auf diese Weise auch die verschiedenen Erscheinungsformen der Maschinenprogramme beschreiben. Je nach dem Zustand der betreffenden, aus elektronischen Bauteilen bestehenden bzw. – bei integrierten Schaltungen – elektronische Bauteile darstellenden Flip-Flop-Speicherregister, der magnetischen Polarisation kleinster Teile von Disketten o.ä. oder dem Fehlen bzw. Vorhandensein von Löchern kann also deren Abfolge als eine Reihe von entsprechenden Symbolen „1“ und „0“, „H“ und „L“ etc. beschrieben werden. Um es noch einmal klar auszudrücken: Das Maschinenprogramm besteht nicht aus diesen Symbolen. Die Verwendung dieser Symbole dient lediglich der grafischen Darstellung von Maschinenprogrammen unter Berücksichtigung des digitalen Charakters von modernen Rechnern. Auch die Darstellung einer elektronischen Schaltung als Zusammenschaltung elektronischer Bauteile besteht in der Verwendung von grafischen Symbolen wie „├—┘“ (für „Widerstand“) und „—■—“ (für „Kondensator“), ohne daß die fertige Schaltung aus diesen Symbolen bestehen würde.

Die Herstellung von Maschinenprogrammen erfolgte in der Frühzeit der Elektronischen Datenverarbeitung tatsächlich durch die Herstellung entsprechender elektrischer Verbindungen zwischen einzelnen elektrischen bzw. elektronischen Bauteilen oder -gruppen innerhalb des Rechners, sei es durch Betätigung entsprechender Schalter, sei es durch Verbindung entsprechender, nach außen geführter Anschlüsse durch Kabel. Selbstverständlich wurden die entsprechenden Verbindungen nicht ad hoc oder aus dem Gedächtnis heraus hergestellt. Die erforderlichen Schalterstellungen bzw. Kabelverbindungen waren in entsprechenden grafischen Darstellungen – Schalt(er)pläne oder Verbindungspläne – fixiert und wurden von dem Bedienungspersonal übertragen, also ausgeführt<sup>22</sup>.

Es ist jedoch grundlegend unrichtig, wenn Clapes, Lynch und Steinberg es so erscheinen lassen möchten, daß diese „Instruktionen für das Umstecken“<sup>23</sup>, also die grafischen – technischen! – Darstellungen, das Maschinenprogramm darstellten. Das Maschinenprogramm ist vielmehr, wie auch ein Laie nunmehr erkennen muß, der durch Betätigen der entsprechenden Schalter bzw. Herstellen entsprechender Kabelverbindungen neu zusammengesetzte Teil des Rechners. Man darf nicht vergessen, daß das Maschinenprogramm nichts anderes ist als der entsprechende konfigurierte Steuerungsteil eines Computers. Die die entsprechenden Schalterstellungen bzw. Kabelverbindungen beschreibende grafische Darstellung hingegen läßt sich ohne Schwierigkeit als der Quellcode erfassen.

Es ist ebenfalls ersichtlich falsch, wenn Clapes, Lynch und Steinberg die Abarbeitung des Programms mit dem Prozeß des Neuverdrahtens gleichsetzen<sup>24</sup>. Die „Abarbeitung der Instruktionen“ (für das Umstecken des Rechners) ist der Prozeß des Neuverdrahtens, die Abarbeitung des Programms hingegen ist der Betrieb des Rechners mit den entspre-

<sup>18</sup> Read Only Memory, auch als Festwertspeicher oder Nur-Lese-Speicher bezeichnet.

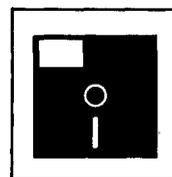
<sup>19</sup> Random Acces Memory = Speicher mit wahlfreiem Zugriff, d.h. jeder beliebige Speicherplatz kann für sich gelesen oder beschrieben werden.

<sup>20</sup> Dies ist nicht zu verwechseln mit der Unterscheidung zwischen statischen RAM's (SRAM's) und dynamischen RAM's (DRAM's). Diese technische Unterscheidung beschreibt lediglich, ob die RAM's ihren Inhalt, also den inneren Schaltzustand, von sich aus allein durch Aufrechterhaltung der Stromversorgung beibehalten, oder ob hierfür zusätzlich eine permanente Auffrischung der Spannungspgcl durch zyklisches Refreshing erforderlich ist, vgl. Dworatschek, Grundlagen der Datenverarbeitung, 8. Aufl., 245f; hierzu König B.III.2., C.I.2.c)bb)(1), C.II.3.b)bb)(2)(a)

<sup>22</sup> Ganzhorn/Walter, Die geschichtliche Entwicklung der Datenverarbeitung, 50f; Dworatschek 26; Koren/Thaller, EDV für jedermann, 25

<sup>23</sup> Clapes/Lynch/Steinberg jur-pc 89, 279, 280

<sup>24</sup> Clapes/Lynch/Steinberg a.a.O.



chend fixierten Schalterstellungen bzw. hergestellten Kabelverbindungen. Nicht die Instruktionen für das Umstecken (die grafische Darstellung) werden bei dem von-Neumannschen Prinzip der Speicherprogrammierung wie Daten in den Hauptspeicher eingelesen, sondern „nur“ deren Ergebnis, wodurch im Hauptspeicher von diesem eine funktionale Kopie<sup>25</sup> erzeugt wird.

Zum besseren Verständnis dieser wichtigen Unterscheidung ist auf die bereits vor Entwicklung der Speicherprogrammierung angewandte und auch später eingesetzte Steuerung durch Lochstreifen hinzuweisen. Schon der deutsche Computerpionier Konrad Zuse benutzte zur Steuerung seiner – allerdings noch elektromechanisch arbeitender – Rechner Z1 bis Z3 in den dreißiger und vierziger Jahren achtspurig gelochte Kinofilmstreifen als Lochstreifen. Diese stellten das Maschinenprogramm dar, denn die Löcher steuerten – abgetastet durch entsprechende mechanische bzw. elektromechanische Fühler – unmittelbar den Rechner<sup>26</sup>. Daraus ergibt sich also, daß bereits vor Entwicklung des Prinzips der Speicherprogrammierung durch von Neumann der „Prozeß der Neuverdrahtung“ mechanisiert war bzw. werden konnte. Bei speicherprogrammierbaren Rechnern wurde der Lochstreifen insgesamt abgetastet und ein elektro-mechanisches bzw. elektronisches Duplikat im internen Speicher hergestellt. Dieses Duplikat stellt die entsprechende konfigurierte Ablaufsteuerung und damit ein Maschinenprogramm dar. Lochstreifenprogramm und rechnerinternes Duplikat sind offensichtlich physikalisch völlig verschieden. Das Lochstreifenprogramm ist ein gelochter Streifen Papier oder Kunststoff; das rechnerinterne Programm besteht aus zusammengeschalteten Relais in unterschiedlichen Schaltzuständen bzw. deren elektronischen Äquivalenten – Magnetkernspeicher, Flip-Flops und deren Miniaturisierungen (RAM). Daher ist für die rechtliche Beurteilung des Maschinenprogramms völlig unerheblich, ob es als Lochstreifen, Lochkarte, Diskette, CD-ROM oder als Halbleiterspeicherelement – wie RAM, ROM, EPROM, Flash-EPROM o.ä. – vorliegt.

Der Begriff „Maschinensprache“ bezeichnet also im Grunde lediglich, welche elektrische Verbindungen zum Hervorrufen bestimmter elektronischer Funktionen vorgenommen werden müssen. Die Verwendung des Wortes „Sprache“ stellt somit lediglich eine Metapher dar<sup>27</sup> und darf keinesfalls zu dem Fehlschluß verleiten, daß hier irgendetwas mit menschlicher Sprache gemeint sei.

Es leuchtet auch Nichtprogrammierern ein, daß das Denken ausschließlich in „Spannung-an“- und „Spannung-aus“-Kategorien eine sehr umständliche und fehlerträchtige Herstellung von Maschinenprogrammen zur Folge haben muß. Zur Vereinfachung wurden jeweils acht dieser „Spannung-an“- oder „Spannung-aus“-Zustände – auch als Bit bezeichnet (binary digit) – zu einer Einheit, einer Vier-Bit-Zahl (Half-Byte oder Nibble), zusammengefaßt und dieser Einheit, deren dezimaler Äquivalenzwert im Dualsystem von 0 bis 15 reicht, das entsprechende Zeichen aus dem Hexadezimalsystem – 0 bis 9 und A bis F – zugeordnet. Ein Byte, also acht Bit, wird demzufolge als eine zweistellige Hexadezimal-Zahl dargestellt<sup>28</sup>.

Da auch der Umgang mit Hexadezimalzahlen nicht gerade zu den einfachsten Dingen im Leben zählt, folgte als nächste Vereinfachung die Zusammenfassung der für eine bestimmte Rechner- bzw. Prozessorfunktion erforderlichen Kombinationen von „Spannung-an“- und „Spannung-aus“-Zuständen – also in der grafischen Darstellung einer Folge von Einsen und Nullen oder sonstigen, unterschiedlichen Symbolen oder einer Gruppe von Hexadezimal-Zahlen – zu Symbolen, die die jeweilige elektronische Funktion symbolisieren. Des besseren und einfacheren „handling“ durch den Menschen wegen wurden diese Symbole der englischen Sprache entnommen. So steht im Entwurf – dem Quellcode – z.B. als Symbol das Kürzel „add“ für die exakt definierte Folge von „Spannung-an“ und „Spannung-aus“ bzw. entsprechenden Schaltzuständen in der Rechnerelektronik – oder aus Löchern in Lochstreifen und Lochkarten, magnetischen Polarisierungen auf Disketten u.ä. –, die im Rechner die elektronische Funktion des Addierens zweier Zahlen zur Folge hat. Diese Abkürzungen werden als „Mnemonics“ bezeichnet; das gesamte System, also die Wahl geeigneter Mnemonics, deren Zuweisung

### Das Lochstreifen-Prinzip

### „Sprache“ als Metapher

### Codierung für „Spannung-an“ und „Spannung-aus“

Binär	Hexa-dezimal	Dezimal
0000	= 0	= 0
0001	= 1	= 1
0010	= 2	= 2
0011	= 3	= 3
0100	= 4	= 4
0101	= 5	= 5
0110	= 6	= 6
0111	= 7	= 7
1000	= 8	= 8
1001	= 9	= 9
1010	= A	= 10
1011	= B	= 11
1100	= C	= 12
1101	= D	= 13
1110	= E	= 14
1111	= F	= 15

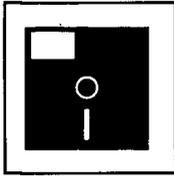
### Mnemonics

<sup>25</sup> Zahmt bezeichnet diese zutreffend auch als „logische Kopie“; Zahmt 106.

<sup>26</sup> Ganzhorn/Walter 45ff; Dworatschek 24; Koren/Thaller 24f; Zabella/Amler, Am Anfang war das Blech, c't 3/98, 232, 234f

<sup>27</sup> vgl. v. Hellfeld, Sind Algorithmen schutzfähig?, GRUR 89, 471; König B.II.3., C.II.3.b)bb)(1)(a)

<sup>28</sup> Dworatschek 59, 78, 144



### *Höhere Programmiersprachen*

zu den einzelnen Prozessor- bzw. Rechnerfunktionen sowie deren Verknüpfungsregeln, heißt *Assemblersprache*<sup>29</sup>.

Es muß noch einmal ausdrücklich daraufhingewiesen werden, daß all diese grafischen Darstellungen, die Folgen von Einsen und Nullen, von Hexadezimal-Zahlen oder von Mnemonics, nicht das Maschinenprogramm, sondern nur einige der möglichen Darstellungsformen, und – wenn nach deren Vorlage das Maschinenprogramm hergestellt wird – auch der Quellcode sind.

Die Herstellung von Maschinenprogrammen erfolgt heutzutage selbstverständlich nicht mehr dadurch, daß ein Mensch den in Einsen und Nullen, Hexadezimal-Zahlen oder Mnemonics entworfenen und aufgeschriebenen Quellcode in einen mechanischen Kodierapparat eingibt, der diese Eingaben in Löcher in Lochstreifen, Lochkarten o.ä. umwandelt. Der Quellcode befindet sich vielmehr maschinenlesbar auf Festplatte oder Diskette und dient als Eingabeinformation bzw. Eingabedaten für das entsprechende Umsetzungsprogramm, den Assembler. Dieser stellt anhand des Quellcodes unmittelbar das auf der Festplatte bzw. Diskette das Maschinenprogramm her.

Eine weitere Vereinfachung erfolgte schließlich durch die Entwicklung der sogenannten höheren oder problemorientierten Programmiersprachen, deren Symbole („Anweisungen“) keine einzelnen Prozessor- bzw. Rechnerfunktionen, sondern eine Vielzahl von Funktionsverkettungen zugewiesen sind. Weitere technische Einzelheiten sind hier entbehrlich. Es genügt zu wissen, daß auch diese Quellcodes als Eingabeinformationen bzw. -daten für ein entsprechendes Programm dienen, das hiernach entweder in toto ein Maschinenprogramm herstellt (Compiler) oder die einzelnen Prozessor- bzw. Rechnerfunktionen unmittelbar und nacheinander auslöst (Interpreter)<sup>30</sup>.

### **Postscript-Belichtung im 24-Std.-Service!**

- Alle DTP-Programme
- MS-DOS und Apple MAC
- Linotype 300/330 Color
- Über 460 Schriften



**Fotosatz Schmidt+Co**  
Heinkelstraße 4  
7056 Weinstadt 3  
☎ 07151-64058,  
Fax 63773

<sup>29</sup> König, Können Objektprogramme urheberrechtlich geschützt sein?, GRUR 89, 559; 561 m.w.N.

<sup>30</sup> König a.a.O.