

Das Urheberrechtsgesetz und Gedanken zu seiner Anwendung

Das Urheberrechtsgesetz soll die Urheber von Werken der Literatur, Kunst und Wissenschaft schützen (§1 UrhG). Allerdings steht dort nichts über Software.

Die geschützten Werke sind laut §2 Abs. 1:

1. Sprachwerke
2. Werke der Musik
3. Pantomimistische und Tanzkunstwerke
4. Werke der bildenden Künste (Baukunst, Kunstwerke)
5. Lichtbildwerke
6. Filmwerke
7. Darstellungen wissenschaftlicher und technischer Art

Programme, also »Computerwerke«, sind noch nicht besonders erwähnt. Wie soll man sie einordnen?

In einzelnen Fällen kann man Grafikprogramme unter Kunstwerke und Musikprogramme unter Musik-

werke einordnen. Den dokumentierten Quellcode eines Programms kann man durchaus als wissenschaftliches Sprachwerk sehen (wegen der verbalen Kommentare).

Wenn Computerwerke Sprachwerke sind, dann sind Programmiersprachen mit Sprachen gleichzusetzen. Da jedoch weder Sprachen wie Englisch und Deutsch noch irgendwelche Dialekte wie Bayerisch und Plattdeutsch Urheberrechtsschutz genießen, wären auch Computersprachen wie Basic, Pascal oder Microsoft-Basic nicht geschützt. Das gleiche gilt für Betriebssysteme wie CP/M oder MS-DOS. Das würde also bedeuten, daß

jeder die Sprachen und Betriebssysteme kopieren und benutzen dürfte, so wie er Esperanto oder Französisch benutzen darf. Das wiederum würde Firmen wie Digital Research und Microsoft die Existenzgrundlage entziehen, da sie doch hauptsächlich vom Verkauf selbstentwickelter Sprachen und Betriebssysteme leben.

Laut Urheberrechtsschutz sind nur »Werke« geschützt, also nur persönliche geistige Schöpfungen (§2 Abs. 2). »Persönlich« besagt, daß es von einer »natürlichen Person«, also einem Menschen, erstellt werden muß. Ein Assemblerquellcode ist also ein Werk. Was aber ist mit dem

Objektcode? Er wurde ja mechanisch mit dem Assembler erzeugt (ASM 2000 eintippen, den Rest macht der Computer und legt das Ergebnis im Speicherbereich \$2000 ab). Auch ist die Übersetzung eines dokumentierten Assemblerquellcodes zum Objektcode keine schützenswerte Bearbeitung im Sinne von §3, denn Bearbeitungen müssen persönliche geistige Schöpfungen des Bearbeiters sein. So gesehen ist nur der Quellcode schutzfähig, und der befindet sich fast nie auf den raubkopierten Disks und auch meistens nicht auf der verkauften Originalsoftware-Disk.

Um nochmal auf die Bearbeitung nach §3 zurückzukommen: Wird ein Programm geknackt, dann ist dies eine persönliche geistige Schöpfung des Knackers, die auch gesondert schützenswert ist, und das laut §3 (wörtlich) »unbeschadet des Urheberrechts am bearbeiteten Werk«. Bearbeite ich also eine Originalsoftware, die eine ganze Diskseite lang ist und Kopierschutz besitzt, so, daß ich ein kurzes, einteiliges Programm habe (eventuell noch mit einem GCS (German Cracking Service) oder Section 8-Vorspann), so ist diese Bearbeitung ohne Beachtung des Urheberrechts am Ursprungswerk auch schützenswert. Auf Deutsch: Ein Knacker darf die Polizei verklagen, da sie ihm seine persönliche geistige Schöpfung beschlagnahmt hat.

Das Ganze läßt sich natürlich noch weiterspinnen: §4 UrhG besagt: »Sammlungen von Werken ... die durch Auslese oder Anordnung eine persönliche geistige Schöpfung sind (Sammelwerke), werden unbeschadet des Urheberrechts an den aufgenommenen Werken wie selbständige Werke geschützt«. Ist also eine kreativ zusammengesetzte Programmsammlung eines Raubkopierers ohne Beachtung des Rechtsschutzes der darin enthaltenen Programme eine eigene geistige Schöpfung?

Das Urheberrechtsgesetz bietet eine Fülle von Paragraphen, mit denen man derartige Gedanken noch bis ins unendliche weiterspinnen kann. Die Entdeckung eines Naturgesetzes, mathematische und logische Gesetze sowie Algorithmen werden nicht vom Urheberrecht abgedeckt, weil sie keine Werke sind. Dazu zählt zum Beispiel der Quicksort-Algorithmus (1962 von C. Hoare entwickelt). Bestehen aber nicht alle Programme aus Algorithmen?

Wenn wir also alles oben Gesagte zusammenfassen, gilt Software (ein

»Computerwerk«) also nur als urheberrechtlich geschützt, wenn es eine persönliche geistige Schöpfung darstellt und die Schutzfrist noch nicht abgelaufen ist. Wegen des Algorithmencharakters muß man jedoch vielen Programmen oder zumindest ihren algorithmischen Teilen (Rechenroutinen, Sortier Routinen etc.) die Schutzfähigkeit absprechen. Das gleiche gilt dann auch für reine Objektcodewerke, das heißt Programme, von denen nur der Objektcode aber nicht zusätzlich der Quellcode veröffentlicht wird. Wie sieht es jetzt mit kopiergeschützten Werken aus?

Ein Verlag, der ein Sprachwerk verlegt, ist aufgrund der Landespressegeseetze verpflichtet, ein Vervielfältigungsstück an die entsprechende Landesbibliothek und ein weiteres Stück an die Deutsche Bibliothek in Frankfurt in lesbarer Form abzuliefern (Pflichtabgabegesetz). Dadurch soll jeder Urheber anhand des hinterlegten Sprachwerks prüfen können, ob sein eigenes Werk in urheberrechtlichem Konflikt zum fremden Werk steht. Dagegen sind Computerwerke oft in doppelter und dreifacher Ausführung unlesbar, insbesondere durch Compilierung, Codierung und Kopierschutzverfahren.

Computerwerke werden somit zu »Geheimwerken«. Sind diese jedoch überhaupt urheberrechtlich schutzfähig?

Nehmen wir als Beispiel ein Goethe-Gedicht, dessen englische Übersetzung dazu und dazu schließlich eine Anzahl von Hexzahlen, die eine Codierung des besagten Gedichts sein soll.

Würde Goethe noch leben, so wäre das Originalwerk noch voll schutzfähig. Die englische Übersetzung ist nach §3 ebenfalls ein voll urheberrechtlich schützenswertes Werk. Was ist aber jetzt mit der Übersetzung in den Code? Angenommen, ein Schwarzkopierer kopiert diese Folge von Hexzahlen, weil er denkt, diesen »ungeschützten Unsinn« kann man ohne Bedenken weiterverbreiten. Daraufhin erstattet der Urheber Anzeige. Der Urheber sagt, es handelt sich um eine Goethe-Übersetzung in einen unknackbaren Code, der Kopierer dagegen behauptet, das Werk sei eine Folge mechanischer Chiffren. Was soll nun der Richter tun, wenn er den Code nicht kennt?

Daraus ist zu schließen, daß bei Geheimwerken (ob codiert oder sonstwie geschützt) nie einwandfrei nachprüfbar ist, ob sie schutzfähig

sind oder nicht. In solchen Fällen kann auch nie jemand aufgrund des Urheberrechtsgesetzes zur Rechenschaft gezogen werden (wohl aber aufgrund anderer Gesetze), denn dann könnte willkürlich jeder behaupten, seine paar Programmzeilen (vielleicht nur zwei PRINT-Befehle) wären ein schützenswertes Kunstwerk.

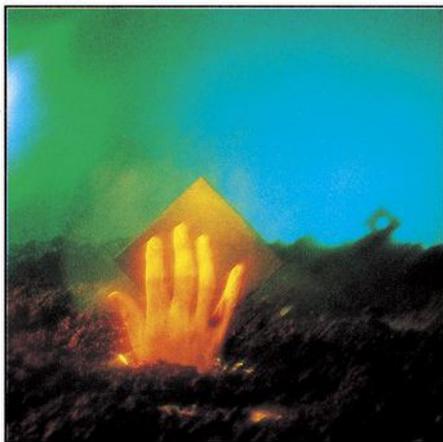
Ein weiterer Grund, der dafür spricht, keine Geheimwerke zu veröffentlichen, ist offensichtlich: Bei vielen Programmen und Programmpaketen kommen immer wieder die gleichen Unterroutinen und Programmteile vor, also Sortier Routinen, Druckroutinen, Routinen für Befehlsweiterungen. Wenn man diese Programme »knackt«, sind oft nicht nur die Leistungen gleich.

Man nehme also die Unterprogramme verschiedener Hersteller, kleide das Ganze in eine neue Bildschirmgestaltung und codiere und schütze das so entstandene Programm so, daß man sich Quellenangaben sparen kann.

Wer Banknoten erhält, muß selbst oder durch Experten prüfen lassen können, ob es Blüten sind. Das Pflichtabgabegesetz verpflichtet Buchverlage, die Werke lesbar zu hinterlegen, so daß auch hier festgestellt werden kann, ob es sich um Plagiate handelt. Bei kopiergeschützter Software ist es jedoch nicht möglich, so etwas festzustellen. Softwarehersteller geben ihre »Geheimwerke« mit großer Selbstverständlichkeit als Urheberwerke aus, obwohl sie es wahrscheinlich oft nicht sind. Daraus würde sich die sinnvolle Empfehlung ergeben, grundsätzlich nur diejenigen Programme unter Urheberrechtsschutz zu stellen, deren Quellcodes veröffentlicht oder öffentlich hinterlegt werden. Doch noch einmal zurück zum privaten Raubkopierer: Wie sieht es denn mit Sicherheitskopien aus?

Nach §53 und §54 UrhG sind einzelne Vervielfältigungsstücke zum persönlichen oder sonstigen eigenen Gebrauch zulässig. Die Kopien (»Werkstücke«) dürfen nicht verbreitet werden, das heißt weder getauscht, noch verkauft, noch verschenkt werden. An Freunde und Bekannte darf man sie allerdings weitergeben, also an Leute, mit denen man durch ein persönliches Band verbunden ist.

Nach einer Antwort des Bundesministers der Justiz vom 16. Juli 1968, auf eine Anfrage und deren Bestätigung durch den Bremer Schulbuchprozeß, darf man bis zu sieben Ver-



Schützer kontra Knackis

**Wie kann man ein Programm sinnvoll schützen?
Und wie kann man solch einen Schutz knacken?
Um diese Fragen zu klären, unterhielten wir uns mit
einigen Programmierern und Knackern.
Sie erklärten uns ausführlich die Grundlagen des Soft-
wareschutzes, des Knackens und des Kopierens.**

vielfältigungsstücke eines Werkes herstellen. Das heißt man darf die Kopie eines Originalprogramms (nicht einer Kopie oder Knackversion) an bis zu sieben Freunde verteilen, die diese aber nicht noch einmal kopieren dürfen. Hier taucht allerdings wieder die Frage auf, was man alles unter den Begriff »Freunde« setzt.

Nehmen wir jetzt aber einmal an, 20 Leute tun sich zusammen, um ein Originalprogramm zu kaufen. Sowohl das Urheberrechtsgesetz als auch bisherige Gerichtsurteile lassen die Frage offen, ob alle zusammen oder jeder von ihnen sieben Kopien für den eigenen Gebrauch herstellen darf. Immerhin wären das 140 legale Kopien..... also keine Raubkopien, sondern »dezentralisierte Sicherheitsbackups«.

Das gesunde Rechtsempfinden eines Richters wird sicher all die oben genannten Überlegungen über den Haufen werfen, aber trotzdem: Das deutsche Urheberrechtsgesetz weist mehr Löcher auf als der Schweizer Käse. Das UrhG schützt nur deutsche Staatsangehörige (§120). Aufgrund des Assimilationsprinzips (Welturheberrechtsabkommen, Artikel II und Berner Übereinkunft, Artikel 3) werden Ausländer im Inland wie Inländer geschützt. Aus diesem Grunde ist auch der amerikanische Copyright Act in der Bundesrepublik nicht anwendbar. Schade, denn dieses Gesetz weist weit weniger Unklarheiten auf: Computerwerke sind darin schon gesondert geregelt.

Und wenn Sie nun sagen, das wäre alles an den Haaren herbeigezogen, dann haben Sie vollkommen recht. Aber sind die Anwälte der Softwarefirmen nicht auch nur dazu da, ihr Recht an den Haaren herbeizuziehen, weil das Gesetz so viele Lücken aufweist?

(M. Kohlen/aa)

Als wir uns nach den Grundlagen der Schutztechniken erkundigten, wurden wir gleich vor eine Frage gestellt: Meinten wir nun den Programm- oder den Kopierschutz? Man klärte uns auf, daß man bei Schutzmethoden folgende Unterscheidung machen muß: Als Programmschutz werden programmtechnische Maßnahmen bezeichnet, die im Computer ablaufen und das Knacken verhindern sollen. Ein Kopierschutz hingegen befindet sich auf der Diskette oder Kassette und soll das Programm vor Kopiersuchen sichern. Nur die Kombination dieser beiden Techniken ist heutzutage zum Schützen von Programmen sinnvoll.

Programmschutz

Ein Programmschutz bedeutet, daß es einem Außenstehenden fast unmöglich gemacht wird, die Arbeitsweise des Programms zu verfolgen. Hier gibt es viele Techniken, die sich allerdings auf die Maschinensprachebene beschränken. Die erste und am weitesten verbreitetste ist die Codierung von Programmteilen. Nur ein kleiner Teil des Programms liegt lauffertig vor, der Rest ist in irgendeiner Form codiert und wird erst bei Bedarf decodiert. Das allein wäre aber nicht effektiv genug: Meistens liegen die Decodierprogramme ebenfalls codiert vor, müssen also von anderen Teilen decodiert werden, und so weiter. Eine Verschachtelung von bis zu zwanzig Codier- und Decodier-Routinen, die sich vielleicht noch gegenseitig aufrufen, ist keine Seltenheit.

Fast immer wird das Codier-Prinzip mit dem Verschiebe-Prinzip gepaart: Die einzelnen Programmteile werden durch Verschieberoutinen über den gesamten Arbeitsspeicher von 64 KByte verstreut. Freaks sagen dazu auch »Spread Code«. Der arme Knacker hat dann praktisch keine Chance, irgendwo einen

Monitor oder ein sonstiges zusätzliches Programm unterzubringen, da in jeder Ecke des Speichers ein paar Byte des Programms stehen. Weiterer Vorteil ist die damit verbundene Reset-Sperre: Man bringt Teile des Programms in Zeropage, Prozessorstack und Bildschirmspeicher unter, denn die werden bei einem Reset, so ausgetüfelt er auch sein mag, immer teilweise gelöscht.

Die nächste Schutzmethode sind die sogenannten »Illegalen Opcodes«, Maschinensprachebefehle, die offiziell gar nicht existieren. Über diese Opcodes haben wir ja schon öfter berichtet. Inzwischen sind die Softwarefirmen von der Verwendung der »Illegals« abgekommen. Aus zwei Gründen: Einerseits kennt die sowieso schon fast jeder, andererseits funktionieren die Opcodes, die man zum Schutz am besten brauchen könnte, nicht auf allen C 64. Der Grund dafür ist, das auch Fremdhersteller die 6510-CPU für Commodore in Lizenz fertigen und daß sich die einzelnen Typen nicht exakt entsprechen. Ungefähr die Hälfte der illegalen Opcodes verhalten sich auf verschiedenen Gruppen von C 64 völlig unterschiedlich. Das bringt natürlich so manchen ausgetüfelten Programmschutz zu Fall, denn was will man mit einem Programm, das nur auf einem Viertel aller C 64 läuft?

Letzte und schwierigste Methode: Selbstmodifizierender Code. Dieser Zungenbrecher bezeichnet nichts weiter, als daß sich ein Programm selbst verändert. In einem Programmteil wandelt sich dann zum Beispiel ein Sprungbefehl nach \$7698 zu einem nach \$4435 um oder ein LDA# wird zu einem STX. Das läßt sich sowohl sinnvoll in Programmen einsetzen, gerade in sehr schnellen Grafik-Routinen, aber auch nur zur Verwirrung des Betrachters. Es geht sogar noch schlimmer: Ein in den Interrupt eingehängtes Programm ändert perio-