

Zauberlehrling

Das Auto232-Protokoll im Detail

Wesentliche Neuerungen im Computerschach des Jahres 1995 waren das Chess232-Brett und der Autoplayer von Christian Donninger. Heute läßt sich der österreichische Schachprogrammierer von den CSS-Lesern (und seinen Kollegen, den Schach-Programmierern) in die Karten sehen, was Funktionsweise und Programmierung dieser Schach-Kommunikatoren betrifft.

„Wir fühlen, daß selbst, wenn alle möglichen wissenschaftlichen Fragen beantwortet sind, unsere Lebensprobleme noch gar nicht berührt sind“ (L. Wittgenstein, Tractatus logico-philosophicus, 6.52).

Auf der letzten Mikro-WM in Paderborn gehörte zur Standardfrage interessierter Besucher „Wie groß ist eigentlich Nimzo3“. Wahrscheinlich hat es mir keiner abgenommen, aber ich wußte und weiß es bis heute nicht. Ganz genau kann ich mich aber noch an Farbe und Größe meines ersten, vor 20 Jahren verfaßten, Programmes erinnern. Vorne war zuerst eine rote Lochkarte, die dem Computer sagte: „Jetzt geht's los“, an zweiter Stelle eine blaue: „Ich bin ein FORTRAN-Programm“. Der dicke Rest weißer, genau einen Dollarschein großer Karten machte das eigentliche Programm aus, und zum Abschluß folgte noch eine grüne Job-Control Karte mit der Botschaft „aus ist's“. Das Ganze paßte ziemlich exakt in eine Adidas Laufschuhschachtel der Größe 42 (das Programm tat, was FORTRAN-Programme so zu tun pflügen, es berechnete physikalische Formeln). Zumindest für Programmierer war damals die Computerwelt noch sinnlicher und bunter.

Wie viele andere Menschen wurde auch ich von meinen ersten (Computer-) Erfahrungen geprägt und stehe daher dem modernen bunten Firlefanz eher reserviert gegenüber. So hatte ich immer das Gefühl, eigentlich kein richtiges Schachprogramm zu schreiben, denn das wäre eines zum Anfassen, mit Brett und Figuren. Als ich einmal Bertold Seifriz von Gambit-Soft mein Leid klagte, meinte er mit echt schwäbischem Pragmatismus „Warum verbindest du nicht die Vorteile beider Welten? Man spielt auf einem Brett, rechnen tut aber der PC“. Und so wurde aus nostalgischem Raunzen ein neues Produkt geboren, das Chess232-Brett, und als natürliche Folge davon der Auto232-Autoplayer.

Wie kommen die Daten in den PC?

PCs werden in der Industrie sehr häufig zum Sammeln von Meßdaten und zum Steuern von Anlagen verwendet. Aus diesem Grund sind sie bereits standardmäßig mit einem entsprechenden Baustein, der seriellen Schnittstelle, ausgestattet (das heute häufigste Meßgerät ist die Maus). Da zum Informationsaustausch mindestens zwei gehören, unterstützen praktisch alle Mikrocontroller (die Rechenhardware der Meßgeräte) ebenfalls diese Übertragungseinrichtung. Beim Chess232-System ist es der gute alte 8051 8-Bit-Controller mit 4 KB Programm und 128 Byte (das ist kein Tippfehler, es sind wirklich 128 Bytes und keine Kilobytes) Datenspeicher. Das Versenden der Daten ist auf Grund der standardisierten seriellen Schnittstelle relativ einfach. Der PC und der Brettcontroller müssen nur mit denselben, einstellbaren Parametern der seriellen Schnittstelle operieren. Vielleicht hat sich der eine oder andere Leser gefragt, was die Endung '232' bei Brett und Autoplayer bedeutet. Im Fachjargon wird die im PC verwendete Schnittstelle als RS-232 bezeichnet.

Eine klassische Figur des Österreichischen Kabarets ist „da Wülde auf seinä Maschin“ (der Wilde auf seinem Motorrad). Sein berühmtester Spruch lautet „I waas zwar net wo i hin wü, aba dafürä bin i schnella durt“ (Ich weiß zwar nicht, wo ich hin will, aber dafür bin ich schneller dort). Es hilft nichts, wenn die Daten dort sind, die andere Seite aber nicht weiß, was damit geschehen soll. Die Daten müssen in einheitlicher Weise interpretiert werden. Zu diesem Zweck hat die Diplomatie schon vor einigen hundert Jahren das Protokoll erfunden. Nachdem sich der Brett-Controller in 128 Byte Speicher nicht nur um die Konversation mit dem PC, sondern auch um die Figurenerkennung, die Steuerung der Leuchtdioden und um die Brett-Tasten kümmern soll, muß ein derartiges Protokoll sehr einfach sein.

Geheimnisvolle Hand

Wie meistens in der Diplomatie, fangen die wahren Probleme erst nach der Lösung der protokollarischen Details an. Letzten Endes müssen die Daten des Bretts bei Genius, Fritz, Rebel, MChess usw. landen und von den Programmen auch richtig verstanden werden. Prinzipiell gibt es dazu zwei Möglichkeiten: Erstens, man überzeugt Richard Lang &

Co. davon, daß künftige Versionen ihrer Programme das Protokoll unterstützen. Die zweite besteht darin, am PC im Hintergrund einen Dolmetscher – im EDV-Jargon auch Treiber genannt – laufen zu lassen, der die Daten des Protokolls übersetzt und den Programmen einen normalen Benutzer vortäuscht.

Die erste Methode hat zwei Vorteile. Man hat selbst keine Arbeit und auch für Lang&Co ist das Ganze jeweils in ca. einem Arbeitstag erledigt. Durch die direkte Kommunikation zwischen Brett und Schachprogramm wird die Wahrscheinlichkeit von „stille Post“-Effekten verringert.

Trotzdem entschied ich mich für die Dolmetscher/Treiber-Methode. Ihr großer Vorteil besteht darin, daß man mit den vorhandenen Programmen spielen kann und nicht erst warten muß, bis die neuesten, mit dem Protokoll ausgestatteten Versionen auf den Markt kommen. Außerdem ging ich davon aus, daß die Brett-Idee in der Luft lag und auch Firmen mit etwas größerer Marktmacht an einem derartigen Produkt arbeiten. Im Falle einer direkten Verbindung hätten diese Firmen auf die Idee kommen können, die Programmierer mehr oder minder sanft zu überreden, das 232-Protokoll doch eher nicht zu unterstützen. Im Gegensatz dazu müßte man schon mit schweren Kanonen auffahren, um den Spatz 232-Treiber zu erlegen.

Täuschungsmanöver

Der 232-Treiber ist ein sogenanntes TSR (Terminate and Stay Resident) Programm. Startet man es, so stellt es zunächst einen DOS-internen Wecker und legt sich im Hauptspeicher schlafen. Das anschließend aktivierte Schachprogramm hat – bis auf den kleinen Schlafplatz – den gesamten Rechner zur Verfügung und wird durch den Treiber nicht gebremst. Der Treiber wird von DOS nur beim Eintreffen von bestimmten Ereignissen geweckt, verrichtet seine meist kurze Arbeit und legt sich anschließend wieder schlafen.

Der 232-Treiber wird entweder durch an der seriellen Schnittstelle ankommende Daten oder durch die Ausgabe von Daten an den Drucker aufgeweckt. Die Daten an der seriellen Schnittstelle sind im 232-Protokoll abgefaßte Züge oder Befehle des Brettes. Der Treiber wandelt diese Information in das Eingabeformat des Programmes (z.B. „e2e4“) um und simuliert anschließend die PC-Tastatur. Drückt man am PC eine Taste, so wird der entsprechende Buchstabe von der Tastatur an einem bestimmten, von MS-DOS reservierten, Speicherplatz geschrieben. Von dort wird er von MS-DOS an ein Anwenderprogramm weitergereicht. Der Treiber übernimmt die Aufgabe der Tastatur und schreibt die Folge „e2e4“ in diesen Bereich. Für das Schachprogramm ist die Treiber-Eingabe von einer eines Benutzers nicht unterscheidbar.

Umgekehrt muß der Treiber natürlich mitbekommen, wann und welchen Zug das Schachprogramm ausführt. Zu diesem Zweck wird zu Beginn des Spieles der „Drucke jeden Zug“ Befehl aktiviert. Der Treiber täuscht dem Schachprogramm vor, ein Drucker zu sein. Wird ein Zug ausgedruckt, weckt MS-DOS zunächst den Treiber auf. Dieser fängt die Daten ab, analysiert sie und wandelt die Druckausgabe in das 232-Protokoll um. Das fertige Protokoll wird anschließend als Paket über die serielle Schnittstelle an das Brett gesandt.

Ein Genius kann nicht Händeschütteln

Nachdem ich den ersten Chess232-Prototyp mit einem Genius-Treiber fertiggestellt hatte, lud ich einen befreundeten Computerschach-Fanatiker ein, um ihm meine neueste Errungenschaft vorzuführen. Wir spielten eine Weile und versuchten – natürlich vergeblich – den Genius1 niederzublitzen. Resignierend seufzte ich „um gegen den 1er Genius zu gewinnen, muß man mindestens ein 2er Genius sein“. Mein Freund schaute mich an und sagte: „Warum eigentlich nicht? Warum ersetzen wir nicht das Brett durch deinen zweiten PC und lassen dort Genius2 laufen?“ An so eine Möglichkeit hatte ich bei der Abfassung des Protokolls überhaupt nicht gedacht.



Das ist
SCHACH
Schach-
Echo
MAGAZIN 64

Aktuell: Erscheint als einzige deutschsprachige Zeitschrift 2x monatlich (24 Ausgaben im Jahr) mit aktuellen Berichten aus aller Welt.

Informativ: Mit 672 großformatigen Seiten pro Jahrgang die umfangreichste deutschsprachige Schachzeitschrift mit Hunderten von Meisterpartien, von Meistern kommentiert, Schachtraining, Theorieartikel.

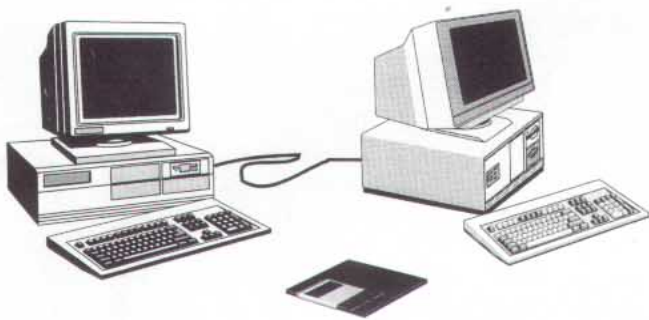
Fordern Sie bitte ausführliche Unterlagen und ein kostenloses Probeheft bei uns an:



VERLAG CARL ED. SCHÜNEMANN KG
Postfach 10 60 67, 28060 Bremen

Aber er hatte recht. Jeder Treiber übersetzt den Zug in das 232-Protokoll und schickt das fertige Paket über die serielle Leitung. Dem Treiber an der anderen Seite ist es vollkommen egal, ob er von einem Brett oder von einem anderen PC einen Zug erhält.

Einzige Klippe war das Händeschütteln. Der Treiber erwartet zu Beginn der Kommunikation mit dem Brett den sogenannten „hand-shake“ Befehl. Erst nachdem sich Brett und PC die Hände geschüttelt haben, kann die eigentliche Kommunikation stattfinden (wie man sieht, haben die Computer-Leute von den Diplomaten tatsächlich einiges gelernt). Das subtile Problem beim Händeschütteln besteht darin, wer die Hand zuerst ausstreckt. Im 232-Protokoll war dazu das Brett vorgesehen. Läßt man zwei Programme gegeneinander spielen, so wartet jedes bis in alle Ewigkeit auf die Hand des anderen.



Einmal in Fahrt, umgingen wir auch dieses Hindernis. Mit Hilfe des Brettes wurde zuerst dem Genius I und dann seinem Nachfolger die Hand geschüttelt. Anschließend ein Druck auf die Alt-M Taste, und es ging wirklich los. Die beiden Programme blitzten gegeneinander. Ich war fasziniert, gleichzeitig fand ich es aber auch gespenstisch. Wir gingen aus dem Zimmer, um uns in der Küche zu stärken. Die Programme spielten ungerührt weiter. Hatte ich am Ende den Menschen im Computerschach abgeschafft?

Herr und Knecht

Nach diesem Versuch war es klar, daß man mit ein paar Modifikationen auch einen allseitig verwendbaren Autoplayer bauen konnte. Etwas subtiler als das Händeschüttelproblem ist das Erkennen des Spiel-Endes. Die meisten Programme drucken nämlich keine Information, wenn eine Partie zu Ende ist. Sie drucken einfach nichts mehr. Die beiden Treiber müssen sich nun darauf einigen, wann ein Spiel zu Ende ist und man folglich ein neues beginnen sollte. Bei der Brett-PC-Kommunikation gibt das Brett nicht nur beim Händeschütteln, sondern auch in diesem Punkt den Takt vor. Es muß also auch im Falle der PC-PC-Verbindung eine Seite den Herrn spielen, der seinem Knecht sagt, „I mecht jetzt spuiln“

(Lass uns ein Spiel machen, Junge), „gnua is“ (Lass es gut sein) und „gemmas wieda an“ (mach weiter, Junge). Der Knecht antwortet darauf immer mit einem folgsamen „jawohl“ bzw., da Computerknechte Pidgin reden, mit einem „okay master“. (Im Computer-Pidgin heißt der Knecht übrigens „slave“). Wer Herr und wer Knecht ist wird durch den Benutzer festgelegt. Der Autoplayer wird durch das Drücken von Alt-0 aktiviert. Jene Seite, auf der diese Tasten gedrückt wurden, ist immer der Herr, das Gegenüber der Knecht.

Wie erkennt nun der Herr das Spielende? Per Definition ist ein Spiel immer zu Ende, wenn eine vom Benutzer einstellbare Anzahl von Zügen überschritten wurde. Üblicherweise tritt dieses Spielende ein, wenn die Programme in einer totremisen Stellung die Figuren sinnlos hin und her schieben. Ein Spiel ist auch zu Ende, wenn über einen bestimmten, ebenfalls einstellbaren, Zeitraum kein Zug mehr gemacht wurde. Drittens kann auch der Benutzer dem Herrn durch einen Tastendruck mitteilen, daß er genug hat.

Die zwei Piepser des Fritz

Da ich ein eingefleischter R. Lang-Fan bin, hatte ich den ersten 232-Treiber für Genius geschrieben. Zu meinem Vergnügen konnte ich feststellen, daß nicht nur das Programm selber, sondern auch die Oberfläche offensichtlich sehr sauber und effizient programmiert ist. In zwei Tagen war der Treiber fertig und funktionierte auch gleich auf Anhieb. Voller Optimismus glaubte ich nun, die Treiber für die übrigen Programme nach Schema F herunterklopfen zu können. Aber nicht jeder ist ein Genius.

Die mit Abstand härteste Nuß war Fritz. Fritz druckte bis Version 3.03 nicht sofort jeden Halbzug, sondern den Zug von Weiß und Schwarz zugleich. Eine aus der Sicht des Benutzers vernünftige Lösung, aus der Sicht des 232-Treibers ein Alptraum. Man konnte nach dem bisherigen Schema nur mit Weiß gegen Fritz spielen. Hat Fritz selbst Weiß, so wartet er zuerst auf den schwarzen Gegenzug, bevor er beide Halbzüge ausdrückt. Der schwarze Gegenzug kommt aber nicht, weil der Brettbenutzer ja noch immer vergeblich auf den weißen Zug von Fritz wartet. Es mußte also eine andere Lösung gefunden werden.

Ich begann den Bildschirmspeicher nach den für jede Figur charakterischen Muster abzusuchen. Der Treiber tastet auf diese Weise das Brett ab und stellt so die Züge fest. Nachdem diese Puzzlearbeit gelöst war, stand noch das Wecker-Problem auf der Tagesordnung. Der Treiber ist ja nicht ständig aktiv, sondern muß durch DOS bei speziellen Ereignissen aufgeweckt werden. Dies ist wiederum nur möglich,

wenn DOS selbst durch einen Befehl des Programmes aktiviert wird. Die Brettdarstellung könnte man zwar theoretisch via DOS vornehmen, allerdings würde dadurch der Aufbau einer neuen Stellung am Bildschirm eine kleine Ewigkeit dauern. Aus diesem Grund umgehen praktisch alle Programme DOS und verändern den Bildschirm direkt. D.h., DOS und somit auch der Treiber haben keine Ahnung, wann Fritz das Brett verändert.

Zum Glück gibt es aber eine „Ton“ Option. Mit dieser Option piepst Fritz bei der Ausführung seiner Züge. Der Piepser wird durch einen DOS-Befehl ausgelöst. Nicht nur der Benutzer, sondern auch der Treiber kann somit durch das Fritz'sche Piepsen auf den Zug aufmerksam gemacht werden.

Ein (in Österreich) berühmter Satz des ehemaligen Bundeskanzlers Sinowatz lautete: „Es ist ja leider alles so kompliziert“. Alle haben darüber gelacht, aber wie recht hatte der Mann. Fritz piepst nämlich auf zwei Arten. Normalerweise führt er den Zug aus und piepst nachher. Wenn der Treiber aufwacht, braucht er in diesem Fall nur noch den Bildschirm zu lesen und die Veränderung festzustellen. Hat Fritz aber die Antwort des Gegners erraten und wird seine Antwort direkt aus dem Permanent Brain gespielt, so wird zuerst gepiepst und dann erst der Bildschirm verändert. In diesem Fall wacht der Trei-

ber zu früh auf, er liest noch den alten Bildschirminhalt und kann keinen Zug feststellen.

FF Neupölla

Auf der unserem Bauernhof gegenüberliegenden Scheune hingen damals ein paar „Einladung zum Bierzelt“-Plakate der FF (Freiwilligen Feuerwehr) Neupölla. Spätestens als das Zwei-Piepser-Problem auftauchte, wurde für mich dieses FF zum Synonym für Fritz Frust, im Lauf der weiteren Arbeiten mutierte es dann zum etwas deftigeren Fu... Fritz. Das Piepser-Problem wurde schließlich gelöst, indem der Treiber nach dem Ton noch für ein paar Sekundenbruchteile wartet und erst danach auf den Fritz-Bildschirm schaut. Fritz hat dann in beiden Fällen den Bildschirm neu gezeichnet.

Voller Stolz schickte ich einem Freund meine neueste Hacker-Leistung. Die Antwort war ernüchternd: „Bei mir geht gar nichts“. Es stellte sich heraus, daß ich immer mit den Standardfarben gearbeitet hatte, er hingegen lieber mit blau eingefärbten schwarzen Feldern spielt. Der Treiber konnte auf blauem Hintergrund die Figuren nicht erkennen. Voller Verzweiflung schickte ich ihm daraufhin die folgende Postkarte:

```
CHES232 FOR X=1 TO 100 FRUST=FRUST+1 IF FRUST =
NERVZUBRUCH*0.9 GOTO LICHTBLICK ELSE NEXT X
LICHTBLICK: STATUS = STATUS + ONELITTLESTEP
CHES232
```

Der Lichtblick war die fritz.par-Datei. In dieser Datei merkt sich Fritz die vom Benutzer gewählten Einstellungen. Ich fand heraus, an welcher Stelle und mit welchen Werten die Brettfarben abgespeichert werden. Der Treiber verändert beim Starten die fritz.par-Datei und stellt auf diese Weise die Standardfarben ein. Nach Spielende werden die vom Benutzer bevorzugten Farben wieder restauriert.

Der Fritz2-Treiber funktioniert noch bis zum heutigen Tag auf diese Weise. Bei Fritz3 schlug allerdings das Sinowatzsche Gesetz erneut zu. Bei jedem Update von 3.00 auf 3.01, 3.02 ... veränderte sich das Format der fritz.par-Datei. Der Treiber veränderte irgendwas, nur nicht mehr die Bildschirmfarben. Inzwischen hatte aber auch das ChessBase-Team von meinen Aktivitäten erfahren. Freundlicherweise stellten sie ab Version 3.04 den Ausdruck auf „Drucke sofort jeden Halbzug“ um. Der neue Fritz-Treiber für diese Version war eine Angelegenheit von wenigen Stunden.

Der tatenlose Rebel

Gideon alias Rebel war beim ersten Test des Treibers nicht sehr beeindruckend. Er tat nämlich gar nichts. Es stellte sich heraus, daß beim Aktivieren

der „Drucke jeden Zug“-Option zunächst der Druckerstatus abgefragt wird. Ist am PC kein Drucker angeschlossen bzw. der Drucker nicht eingeschaltet, wird die Aktivierung dieser Option verweigert. Der Treiber tritt einen langen Schlaf an. (Andere Programme sind weniger kleinlich. Sie schicken die Daten so oder so an den Drucker. Ist kein Drucker vorhanden bzw. nicht eingeschaltet, hat das keine Auswirkungen.)

Die Lösung dieses Problems ist weniger kompliziert. Der Treiber fängt auch die Druckerstatus-Abfrage ab und meldet Gideon/Rebel „Drucker eingeschaltet, alles in Ordnung“. Trotzdem sollte auch in diesem Fall der Sinowatzsche Satz seine Gültigkeit behalten. Bei Rebel7 taucht dieses Problem plötzlich wieder auf. Ich habe daraufhin das Rebel-Team angerufen und gefragt, was sie denn nun anders machen. Sie schwören Stein und Bein, in diesem Bereich den Code von Rebel6 nicht angerührt zu haben. Gemeinerweise kann ich auch auf meinen PCs ohne die geringsten Probleme mit Rebel7 spielen.

Es ist wirklich sehr kompliziert (unmöglich), einen Fehler zu finden, der auf der eigenen Anlage gar nicht auftritt. Die einfachste Lösung des Rebel7-Problems besteht darin, den Rebel auf jenem PC spielen zu lassen, an dem der Drucker angeschlossen ist (der Drucker muß natürlich auch eingeschaltet werden).

Tals Probleme mit der Umwandlung

Ein im Durchschnitt nicht sehr starkes, dennoch aber interessant-exzentrisches Programm ist das Chess System Tal. Meistens spielt es mäßig, im Endspiel katastrophal (siehe etwa Nimzo3 gegen CS-Tal, Mikro-WM 95), aber ab und zu nimmt es einen Anlauf in Richtung Genialität (Genius3 gegen CS-Tal, ebenda). Das Tal-Team bat mich, für eine Vorabversion einen Treiber zu schreiben. Nachdem das Chess System in derselben Weise wie der Genius3 druckt, bastelte ich diesmal wirklich nach Schema F im Handumdrehen einen Treiber. Bereits beim ersten Versuch schien alles glatt zu gehen. Leider war aber Nimzo3 von Tals Mattangriffskünsten nicht sehr beeindruckt, es wickelte in Tals Selbstmordphase, das Endspiel, ab. Die Bauern wurden mit tatkräftiger Mithilfe Tals nach vorne vorgeschoben. Bei der ersten Bauernumwandlung hatte das Chess System genug, es stürzte mit der Meldung „Illegal Promotion“ ab. Inzwischen ist dieser nette Fehler behoben, das Chess System aber – beim Abfassen dieser Zeilen – noch immer nicht auf dem Markt. Vielleicht ist Chris Whittington doch zur Erkenntnis gelangt, daß Tal die anderen nicht immer bereits vor dem Endspiel Matt setzt.

Ruhestand des Dolmetschers

Das Auto232-System hat sich als Quasi-Standard durchgesetzt. Die Schwedische Eloliste und das Welser Turnier werden bereits weitgehend automatisch abgewickelt. Ich habe auch einige Anfragen von Autoren weniger bekannter Programme erhalten, ob ich nicht für sie Treiber schreiben könnte. Ich sage ihnen immer freundlich, aber bestimmt ab. Es zahlt sich einfach nicht aus, mein Stundenlohn würde – die Betriebskosten für PC und Büro noch gar nicht gerechnet – ca. 2 DM betragen. Bei Windows-Programmen funktioniert außerdem die Hand im Hintergrund nicht mehr auf dieselbe Manier.

Nachdem der Autoplayer auch den einzelnen Programmierern Vorteile bringt (es war noch nie so leicht, Testspiele gegen alle wichtigen Programme durchzuführen), schlage ich eine Aufteilung der Lasten vor. Auf der Service-Diskette zu dieser CSS-Ausgabe wird das 232-Protokoll veröffentlicht. Jeder kann und soll dieses Protokoll in seinem eigenen Programm verwenden. Ich erspare mir das Schreiben von Dutzenden neuer Treiber, und jeder kann mit jedem spielen.

Die Geister, die er rief

Auf der Mikro-WM in Paderborn wurde von einem Programmierer übrigens der Vorschlag gemacht, mit dem Autoplayer zu spielen. Ich bin ein entschiedener Gegner dieser Idee. Im ersten Schritt spielen die Programme miteinander, die Programmierer schauen zu. Bald werden einige daraufkommen, daß es billiger und einfacher ist, zu Hause zu bleiben. Alt-0 drücken kann schließlich auch der Turnierdirektor. Zum Schluß bleibt nur noch ein einsamer Turnierdirektor übrig, der alle paar Stunden den Spielstand überprüft. In Wels (siehe Nachrichten) ist diese Fiktion bereits weitgehend Realität.

Den größten Teil des Jahres sitzen die Schachprogrammierer im stillen Kämmerlein alleine vor ihren Kästen. Diese Turniere sind für mich so etwas wie die Dorffeste der Computerschach-Gemeinde. Man trifft sich, tauscht seine neuesten Ideen aus, trinkt ein Bier miteinander, manche führen ihre jahrelang sorgsam gepflegten Fehden weiter. Auch wenn ich dann nach einer Woche für einige Zeit wieder genug Trubel genossen habe, freue ich mich trotzdem jedesmal schon wieder auf den nächsten Kirchtag. Ich hoffe, nicht der Zauberlehrling zu sein, der eines Tages die Geister verflucht, die er selbst geschaffen hat.

Der Autoplayer AUTO232 von Chrilly Donninger wird exklusiv vertrieben durch CSS-Shop, Entenweg 34, D-22549 Hamburg, Tel. 040/83293133.
