

```

*****
*
*   Datensicherung auf USB-Stick mit Hilfe eines Batchprogramms
*   =====
*
*   Eine Schritt-für-Schritt-Anleitung für Einsteiger in die
*   -----
*
*               Batchprogrammierung
*               -----
*
*           Datei: BATHTIP.TXT
*
*           Autor: Dr. Georg Niepel
*                  Kaiserplatz 11
*                  52222 Stolberg
*
*           Datum: Okt. 2006
*
*****

```

Lieber Leser,

mit Hilfe dieser Anleitung soll es Ihnen möglich sein, in kurzer Zeit kleine BATCH-Programme zu schreiben, die funktionieren und Ihnen bei Ihrer Verwendung Ihres Computers praktische Dienste leisten.

Ein erstes Ziel soll ein BATCH-Programm sein, dass Ihnen auf einen Knopfdruck die Datensicherung aller Ihrer wertvollen Daten erlaubt. Natürlich benötigen Sie irgend einen dafür brauchbaren Massenspeicher. In dieser Anleitung wird dafür ein RAM-Stick verwendet, der eine Reihe von Vorteilen bietet. Wer will, kann - evtl. mit kleinen Änderungen am Programm - auch andere Speichermedien verwenden.

BATCH ist auf Deutsch "Stapelverarbeitung". Damit ist gemeint, dass eine Reihe von Befehlen (der sogenannte Stapel) nacheinander abgearbeitet werden. Es werden dabei DOS-Befehle verwendet.

BATCH ist von Microsoft ursprünglich für DOS vorgesehen worden. Im Prinzip funktionieren DOS-Befehle aber auf jedem WINDOWS-Rechner. Deswegen können wir die Vorteile, die BATCH bietet, auch für Ihren Rechner nutzbar machen, wobei es ziemlich egal ist, welche WINDOWS-Version Sie gerade benutzen.

Wobei wir bei einem heiklen Punkt angekommen wären: Meine BATCH-Programme wurden auf WIN98SE, auf XP Home und auf XP Professional erprobt. Dabei habe ich bemerken müssen, dass manche DOS-Befehle je nach WINDOWS-Version minimal unterschiedlich funktionieren. Das muss nicht stören. Sollte etwas auf Ihrem Rechner überhaupt nicht oder nicht ordentlich funktionieren, dann ärgern Sie sich bitte nicht. Probieren hift hier meistens weiter.

Fangen wir an. Unser Programm soll BACKUP heissen. Wir benötigen als erstes einen File, in den wir unser Programm schreiben. Dazu müssen wir einen File eröffnen. Tun Sie dies am besten von WINDOWS aus, denn mit WINDOWS haben Sie es einfacher.

Sie wissen nicht, was File ist? Es handelt sich um eine Datei. Hier wird es eine sogenannte BATCH-Datei sein. Eine BATCH-Datei wird im Prinzip wie eine Textdatei behandelt, nur dass der Dateiname nicht mit .TXT endet, sondern mit .BAT. Dazu aber später.

Vorher überlegen wir noch kurz, in welchen Ordner wir den File ablegen wollen. Ich schlage vor, dass Sie im Hauptverzeichnis C:\ einen Ordner eröffnen, den Sie SKRIPT nennen.

Sie wissen nicht, wie Sie das machen sollen? Doppelklicken Sie auf

Arbeitsplatz. Doppelklicken Sie nun auf Festplatte (C:), das meist als lokales Laufwerk bezeichnet wird. Halten Sie den Mauszeiger innerhalb des Fensters ins Leere und klicken Sie die rechte Maustaste. Es wird ein Menü aufgeblättert, in dem auch "neu" vorkommt. Dort zeigen Sie mit dem Mauszeiger drauf. Es blättert sich wiederum ein weiteres Menü auf, in dem ganz oben "Ordner" steht. Dort klicken Sie mit der linken Maustaste hin. Sofort sehen Sie - blau unterlegt - einen neuen Ordner. Sie sollten den Namen "neuer Ordner" natürlich sofort in den von uns gewählten Namen SKRIPT umändern, um ihn dann zu öffnen.

Sie wissen nicht, wie man einen Ordner umbenennt? Au weia, es wäre gut, Sie lernen erst einmal, mit WINDOWS umzugehen, bevor Sie hier weitermachen. Niemand will Sie frustrieren, aber so kommen wir nicht weiter. Umbenennen geht mit Rechtsklick auf den Ordernamen. Den Ordner nennen wir also Skript. Durch Doppelklicken öffnen wir ihn.

In den leeren Ordner SKRIPT plazieren wir nun die BATCH-Datei. Dazu legen wir eine .TXT-Datei an und nennen diese dann BACKUP.BAT.

Sie wissen nicht, wie das geht? Rechtsklicken Sie in den leeren Ordner. Im Menü - sie kennen das schon - zeigen Sie auf "Neu". Es blättert sich ein Menü auf - Sie erkennen es wieder - aber diesmal wählen Sie eine Textdatei. Beim Umbenennen der Textdatei schreiben wir

BACKUP.BAT

Es ist sehr wichtig, aber hier muss jeder Buchstabe exakt stimmen. Windows kommt mit einer Warnung, dass wir doch bitte nicht die Dateinamenerweiterung ändern sollen usw. Das ignorieren wir aber und antworten, dass wir dennoch die Änderung durchführen wollen.

Wir haben jetzt eine Datei namens BACKUP.BAT, und diese Datei findet sich im Ordner SKRIPT, und dieser Ordner ist im Hauptverzeichnis. Für den geübten Benutzer von WINDOWS ist es kein Geheimnis, dass man also die Datei mit voller Pfadbezeichnung wie folgt angibt:

C:\SKRIPT\BACKUP.BAT

Für den Anfänger: Der Buchstabe am Anfang bezeichnet das Laufwerk, in diesem Fall steht C: für das lokale Laufwerk, die Festplatte. Man darf den Doppelpunkt nicht vergessen. Der Schrägstrich dahinter (bitte beachten Sie, dass der Schrägstrich nach links kippt) wird zur einheitlichen Verdeutlichung besser "backslash" genannt. Sagen Sie bloß, Sie finden das Zeichen auf Ihrer Tastatur nicht!! Halten Sie die "AltGr" Taste gedrückt und betätigen Sie dann ß (und reißen Sie mir nicht den Kopf ab, wenn backslash auf Ihrer Tastatur woanders steht).

Backslash zeigt in der Pfadbezeichnung an, dass als nächstes ein untergeordneter Ordner folgt (oder, beim letzten backslash, dass der Dateiname folgt). Der Name der Datei endet mit einem Punkt und drei nachfolgenden Buchstaben, die man Dateierweiterung nennt.

Wie Sie schon bemerkt haben, steht .BAT für BATCH-Datei. Der Rechner weiß aufgrund der Dateierweiterung .BAT, dass er hier eine Stapelverarbeitung vor sich hat. Wenn das BATCH-Programm schließlich einmal fertig sein wird (gegen Ende dieser Erläuterungen), dann kann man den Rechner auffordern, die Stapelverarbeitung zu starten, indem man einfach auf den Dateinamen BACKUP.BAT doppelklickt.

Wir müssen natürlich das BATCH-Programm erst noch schreiben. Dafür - für das Schreiben dieser BATCH-Datei - können wir NICHT AUF DEN DATEINAMEN DOPPELKlicken. Wir müssen auf den Dateinamen BACKUP.BAT Rechtsklicken und in dem sich dann öffnenden Menü auf Bearbeiten klicken. Jetzt können wir unsere BATCH-Datei schreiben.

Wie immer beim Programmieren, muss man die Befehle kennen, die man benutzen kann. Wir werden hier ein paar DOS- bzw. BATCH-Befehle kennen lernen. Was wir später darüber hinaus noch benötigen, können wir uns im FASTHELP oder im DOSHELP heraussuchen. Dabei handelt es sich um Programme bzw. Dateien, die den späten DOS-Versionen und den frühen WINDOWS-Versionen beigelegt waren. Suchen Sie mal in Ihrem Rechner danach. Oder Sie kennen jemanden, der Ihnen weiterhilft. Oder es geht Ihnen wie mir, der sich mühsam alles zusammensuchen musste und viele Dinge bis heute nicht weiss. Wenn es Ihnen auch so gehen sollte, dann ärgern Sie sich nicht; es ist eines der Grundprobleme von Microsoft (MS), dass MS die Dokumentation sträflich vernachlässigt und die vorhandene Dokumentation so verquast schreibt, dass sie kein Mensch richtig versteht.

Fast alle BATCH-Dateien (BATCH-Programme) fangen mit dem Befehl

```
@echo off
```

an. Das hat damit zu tun, dass Sie sonst - ohne diesen Befehl - bei der Ausführung des BATCH-Programms jeden ausgeführten BATCH-Befehl angezeigt bekommen, was Sie bestimmt nicht wollen, denn es schreibt Ihnen den Bildschirm voll mit den Befehlen, von denen Sie vielleicht froh sind, dass Sie sie nicht mehr sehen müssen, nachdem Sie sich einmal durch das Programmieren hindurchgequält haben. Tippen wir also den Befehl `@echo off` in die erste Zeile.

Wichtiger Hinweis: Sie sollten von vorne herein immer dafür sorgen, dass nirgendwo ein Leerzeichen zu viel oder zu wenig steht, dass jeder Buchstabe und jedes Zeichen fehlerfrei sind. Sonst wird die BATCH-Datei nicht ausführbar sein, d.h. wenn man das Programm ausführen will, dann werden ERROR-Meldungen erfolgen. Später den Fehler zu finden kann Stunden, Schweiß und Tränen kosten.

Der nächste Befehl in unserem BATCH-Programm soll dafür sorgen, dass eine Begrüßung in die Anzeige geschrieben wird. Wir schreiben also:

```
echo -----  
echo      BACKUP-Programm  
echo      Datensicherung  
echo      auf dem RAM-Stick  
echo -----
```

Diese fünf Befehlszeilen sorgen dafür, dass fünf Zeilen in die Anzeige geschrieben werden, wenn das BATCH-Programm startet, nämlich die aus fünf Zeilen bestehende Begrüßung. Dabei sorgt der Befehl `echo` dafür, dass der nach dem Leerzeichen folgende Text auf den Bildschirm geschrieben wird. Wichtig ist das Leerzeichen nach dem Befehl `echo`. Sie können, wenn Sie wollen, selber einen anderen Text verwenden, Ihren Namen angeben oder die Bezeichnung für den Rechner angeben, für den Sie dieses Programm vorgesehen haben.

Es folgen jetzt weitere Befehle. Mit den zwei Befehlszeilen

```
echo.  
echo.
```

sorgen wir dafür, dass zwei Leerzeilen in die Anzeige geschrieben werden. Dies ist aus optischen Gründen ganz schön, weil als nächstes die Aufforderung folgen soll, mit Tastendruck auf eine beliebige Taste das Programm fortzusetzen. Der dazugehörige Befehl lautet

```
pause
```

und diesen Befehl tippen wir nun auch in unser BATCH-Programm. Es sollte jetzt so aussehen:

```

@echo off
echo -----
echo          BACKUP- Programm
echo          Datensicherung
echo          auf RAM-Stick
echo -----
echo.
echo.
pause

```

Ich empfehle Ihnen, nach diesem ersten Teilabschnitt zu speichern, damit wir unsere bisherige Arbeit ausprobieren können. Zum Speichern: Oben links das Menü "Datei" anklicken, dann "Speichern". Nun schließen wir die Datei (das Kreuzchen oben rechts anklicken). Jetzt können wir durch Doppelklick auf "BACKUP.BAT" unsere bisherige Arbeit begutachten. Wir sehen, dass sich ein DOS-Fenster öffnet. Das DOS-Fenster ist dadurch gekennzeichnet, dass wir weiße Zeichen auf schwarzem Grund sehen. In diesem DOS-Fenster erscheint unser Begrüßungstext und die Aufforderung, eine beliebige Taste zu drücken. Hat bisher alles geklappt? Na prima!

Wenn nicht, sollten Sie unbedingt die Datei BACKUP.BAT noch einmal durch Rechtsklicken/Bearbeiten öffnen und Zeile für Zeile nachprüfen, ob Sie sich nicht verschrieben haben. Außerdem sollten Sie nachprüfen, ob der Dateiname mit .BAT endet.

Es geht weiter mit dem BATCH-Programm. Im jetzt folgenden Programm-Abschnitt wollen wir prüfen, ob der RAM-Stick im WINDOWS-System als Laufwerk E: angemeldet ist, also funktionsfähig ist. Dazu wenden wir einen kleinen Trick an. Wir sehen eine Textdatei vor, die wir

BACKUP.TXT

nennen wollen, und streben an, diese auf den RAM-Stick zu speichern. Im Anschluss daran überprüfen wir, ob sich die Datei auf E: finden läßt.

Zu Beginn wollen wir also die Textdatei erzeugen und abspeichern. Im Ordner Skript öffnen wir eine neue Textdatei (Dateiendung .TXT) und umbenennen diese auf den Namen BACKUP.TXT. Dann öffnen wir die Datei und schreiben einen Text hinein. Ich schlage den Text vor: "Bitte diese Datei nicht löschen". Dann speichern wir die Datei. Die Datei hat also den Pfad:

C: \Skript\BACKUP.TXT

Jetzt setzen wir die Programmierung fort. Wenn nicht schon geschehen, öffnen wir die Datei BACKUP.BAT wieder mit Rechtsklick/Bearbeiten und geben weitere Befehlszeilen ein:

```
if not exist E:\BACKUP.TXT COPY C:\SKRIPT\BACKUP.TXT E:\BACKUP.TXT
```

Was bedeutet das? Wie der Befehl "if not exist" in seiner englischen Bedeutung schon andeutet: "falls die folgende Datei nicht vorhanden ist (nämlich die Datei E:\BACKUP.TXT) soll der dahinter folgende Befehl ausgeführt werden (der lautet copy; eine Datei soll kopiert werden)." Es wird also überprüft, ob auf dem RAM-Stick E: bereits die Datei BACKUP.TXT vorhanden ist. Dies könnte nur sein, wenn schon in der Vergangenheit einmal die Datei auf den Stick gespeichert und dann nicht gelöscht worden wäre. Also: Ist bereits auf E: die Datei vorhanden? Falls ja, geschieht gar nichts. Falls nein, wird der nachfolgende Befehl COPY ausgeführt.

Der Copy-Befehl führt einen Kopiervorgang aus. D. h. eine Datei wird

von der einen Stelle (Quelle) zur anderen Stelle (Ziel) kopiert. Die Datenquelle wird hinter dem Copy-Befehl zuerst angegeben (hier: C:\Skript\BACKUP.TXT), das Ziel wird nach einem Leerzeichen dahinter angegeben (hier: E:\BACKUP.TXT).

Im Ganzen: Wenn auf dem RAM-Stick die kleine Textdatei noch nicht vorhanden ist, soll diese Textdatei vom Rechner geholt werden und auf den RAM-Stick rüberkopiert werden. Soweit verstanden? Lesen Sie jetzt die Befehlszeile noch einmal und verstehen Sie sie so:

```
if not exist (falls nicht existiert) E:\BACKUP.TXT
(auf dem Laufwerk E: die Datei BACKUP.TXT, dann soll der folgende
Befehl ausgeführt werden, nämlich) Copy (kopiere die folgende Datei)
C:\Skript\BACKUP.TXT (nach dem RAM-Stick unter dem Namen)
E:\BACKUP.TXT
```

Es geht schon! Man muss sich nur etwas hineindenken.

Jetzt folgt weiter im Programm die Abfrage, ob die Datei nun inzwischen auf dem RAM-Stick auslesbar ist. Man stelle sich vor: Ist der RAM-Stick defekt oder nicht vorhanden, dann findet DOS auch keine Datei. Andernfalls, wenn der RAM-Stick intakt ist, müsste nach dem Kopiervorgang ja nun die Datei BACKUP.TXT vorhanden sein. Also schreiben wir als nächste Programmzeile:

```
if not exist E:\BACKUP.TXT echo !!! Laufwerk E: arbeitet nicht !!!
```

Diese Zeile können Sie jetzt schon verstehen: Falls die Datei E:\BACKUP.TXT nicht existiert, dann soll auf den Bildschirm geschrieben werden:

```
!!! Laufwerk E: arbeitet nicht !!!
```

Denn wenn Laufwerk E: arbeiten würde, dann müsste ja diese Datei vorhanden sein.

Nachdem Sie die letzten Programmzeilen durchschaut haben, werden Sie die nächsten beiden Programmzeilen ganz leicht verstehen. Sie haben das Ziel, bei defektem RAM-Stick des Benutzer zu informieren und das Programm dann zu beenden:

```
if not exist E:\BACKUP.TXT echo !!! keine Datensicherung !!!
if not exist E:\BACKUP.TXT exit
```

Wie Sie sich schon gedacht haben, bedeutet `exit` Beendigung des BATCH-Programms.

Wir wollen natürlich davon ausgehen, dass der RAM-Stick arbeitet, also müssen wir für diesen Fall auch etwas vorsehen. Die folgende Programmzeile informiert den Benutzer, wenn der RAM-Stick in Ordnung ist:

```
if exist E:\BACKUP.TXT echo !!! RAM-Stick arbeitet !!!
```

Der Befehl `if exist` bedeutet also "wenn die Datei existiert". Das war doch noch ganz einfach, oder? Eigentlich geht es so einfach weiter.

Denn nun wollen wir den kleinen Textfile wieder löschen. Er hatte ja nur die Aufgabe, uns die Überprüfung zu gestatten, ob das Laufwerk E: arbeitet. Nachdem wir das nun wissen, ist die kleine Datei ja jetzt überflüssig und kann vom RAM-Stick gelöscht werden. Natürlich soll dieser Textfile nur auf dem RAM-Stick gelöscht werden; auf unserem PC bleibt dieser File für's nächste Mal natürlich stehen. Das Löschen geschieht gezielt auf dem RAM-Stick (Laufwerk E:).

Dazu dient die Programmzeile:

```
DEL E:\BACKUP.TXT
```

Hier bedeutet DEL einfach "delete", d.h. lösche. Was nach DEL folgt, ist ein Leerzeichen, und dann der Pfad zu der Datei, die gelöscht werden soll. Es folgt jetzt eine extrem wichtige Warnung: FINGER WEG FÜR ALLE ANFÄNGER UND ALLE LEUTE, DIE NICHT WISSEN WAS SIE TUN. Denn ein mit DEL gelöschter Datensatz ist ohne jede weitere Warnung ein für alle Male gelöscht und weg. Das macht bei unseren selber geschriebenen Dateien nicht so viel aus, aber wehe, wir löschen wegen eines Tippfehlers vielleicht eine wichtige Datei im Rechner...! Also passen Sie auf, dass diese Zeile ohne Fehler geschrieben wird. Oder, was notfalls auch geht, lassen Sie die Zeile einfach weg. Es schadet nichts, wenn die kleine Datei einfach auf dem RAM-Stick stehen bleibt.

Weiter gehts mit einer erneuten Erprobung unserer bisherigen Arbeit. Wie das geht, wissen wir noch vom letzten Mal. Wir können erwarten, dass der Bildschirm anzeigt, ob der RAM-Stick arbeitet. Wenn das kleine BATCH-Programm abgelaufen ist, können wir überprüfen, indem wir Windows-Explorer benutzen, ob der Löschvorgang auf dem RAM-Stick ebenfalls geklappt hat.

Jetzt soll wieder weiter programmiert werden. Das nächste Ziel ist etwas anspruchsvoller. Wir wollen Datum und Uhrzeit der Datensicherung auf dem RAM-Stick hinterlegen, und zwar in Form einer Datei, die wir DATESTMP.TXT nennen wollen. Der Name soll "date stamp" bedeuten. (Datum-Zeit-Stempel).

Soweit ich weiß, ist BATCH mit dieser Aufgabe überfordert. D.h., wir müssen jetzt BATCH verlassen und diese Aufgabe mit QBASIC erledigen. Keine Sorge! So schwierig ist das nicht.

Das Programm DATESTMP.BAS soll im Ordner BASIC vorliegen, und der Ordner BASIC befindet sich bei mir unter dem Pfad

```
C:\Programme\BASIC\
```

An dieser Stelle folgt ein kleiner Excurs zu den Pfadbezeichnungen unter DOS. Leider hat DOS seine Probleme bei allen Datei- und Pfadbezeichnungen, die länger sind als 8 Zeichen. Deswegen muss die Bezeichnung "Programme", die wir von WINDOWS kennen, in DOS anders genannt werden. Immerhin, es geht! Das ist ja schon mal was wert. Also: unter DOS heißt es nicht "C:\Programme\" sondern

```
C:\PROGRA~1\
```

Sehen Sie genau hin! Nach den 6 Zeichen PROGRA folgt das Zeichen ~ und danach die Ordnungszahl 1.

Alle Ordner- und Dateibezeichnungen, die mit PROGRA anfangen, werden durchnummeriert! Glücklicherweise gibt es unter WINDOWS nicht so viele Ordnernamen, die mit PROGRA anfangen, so dass wir uns darauf verlassen können, dass wir mit PROGRA~1 immer richtig liegen.

Zurück zu unserem Plan, QBASIC einzusetzen. Der Pfad unseres geplanten QBASIC-Programms lautet:

```
C:\PROGRA~1\BASIC\DATESTMP.BAS
```

Natürlich weiß ich nicht, ob Sie in Ihrem Rechner die Dateistruktur auch so aufgebaut haben, oder eventuell anders. Hier können Sie sich natürlich nach dem richten, wie Sie es in Ihrem Rechner vorfinden. Ich schlage Ihnen aber aus gutem Grund vor, für QBASIC-Programme und für

BATCH-Programme ganz kurze Pfade aufzubauen, damit Sie es nicht unnötig schwer haben. Es ist doch ganz einfach, wenn im Hauptverzeichnis Ihres Rechners ein Ordner "Programme" existiert, in diesem Ordner einen Unterordner "BASIC" anzulegen. Oder sie legen vielleicht direkt im Hauptordner einen Ordner BASIC an. Machen Sie es so, wie es zur Dateistruktur Ihres Rechners am besten passt.

In diesem Ordner BASIC müssen alle notwendigen Dateien von QBASIC vorhanden sein, sonst läuft QBASIC nicht oder nicht richtig. Zu diesen Dateien gehören:

EDIT.COM
EDIT.HLP
QBASIC.EXE
QBASIC.HLP
QBASIC.INI

Die für QBASIC nötigen Dateien finden Sie in höheren DOS-Versionen oder in den ersten WINDOWS-Versionen. Oder fragen Sie mal jemanden... In diesem BASIC-Ordner soll auch unser QBASIC-Programm mit Namen DATESTMP.BAS zu finden sein.

Lassen Sie uns an dieser Stelle davon ausgehen, dass das QBASIC-Programm unter dem Pfad C:\PROGRA~1\BASIC\DATESTMP.BAS zu finden ist, und dass dieses Programm einen kleinen Textfile erzeugt, der aktuelles Datum und aktuelle Uhrzeit enthält. Der Pfad soll lauten:

C:\Skript\DATESTMP.TXT

Das QBASIC-Programm soll an anderer Stelle beschrieben werden. Lesen Sie am besten die Kommentarzeilen im QBASIC-Programm. Hier geht es jetzt um die BATCH-Programmierung.

Um das BASIC-Programm zu starten, formulieren wir die nächste BATCH-Programm-Zeile wie folgt:

C:\PROGRA~1\BASIC\QBASIC /RUN C:\PROGRA~1\BASIC\DATESTMP.BAS

Am Anfang der Zeile wird QBASIC aufgerufen, und zwar sicherheits- halber mit der ganzen Pfadlänge. Wir gehen dadurch sicher, dass Ihr Rechner QBASIC auf jeden Fall findet. Natürlich gilt die Pfadangabe nur, wenn auch Sie auf Ihrem Rechner das QBASIC unter diesem Pfad vorhalten. Sollten Sie in Ihrem Rechner das QBASIC woanders abgespei- chert haben, so müssen Sie den bei ihnen richtigen Pfad angeben. Hinter dem Leerzeichen (wichtig!) folgt der Befehlsanhang /RUN Damit wird QBASIC der Befehl gegeben, das nachfolgend angegebene Programm auszuführen. Dieses lautet: C:\PROGRA~1\BASIC\DATESTMP.BAS Auch hier gilt: Sie müssen den Pfad ggf. an die Verhältnisse in Ihrem Rechner anpassen.

Die nächste Befehlszeile dient dazu, den Datum-Zeit-Stempel, der ja als kleine Textdatei mit dem Pfad C:\Skript\DATESTMP.TXT vorliegt, auf dem RAM-Stick zu kopieren, damit unsere Datensicherung auf diese Art datiert werden kann. Den Copy-Befehl kennen wir schon:

copy C:\Skript\DATESTMP.TXT E:\DATESTMP.TXT

Wir sehen: Die kleine Datei mit dem Datum-Zeit-Stempel wird direkt auf das Laufwerk E: kopiert, und heisst dort genauso (DATESTMP.TXT).

Jetzt wollen wir den Benutzer wissen lassen, dass dieser Vorgang erfolgreich absolviert ist. Dazu dient diese Befehlszeile:
echo Das Datum der Datensicherung wurde gespeichert
Hierzu gibt es nichts mehr zu erklären, glaube ich.

Und nun soll das eigentliche Kopieren der Dateien beginnen.
Dazu die Ankündigung:

```
echo ...Daten-Backup beginnt ...
```

Und dann die Kopierbefehle. Wer einen sehr großen Massenspeicher besitzt, kann ja einmal eine komplette Datensicherung des gesamten Rechners versuchen. Der Befehl lautet:

```
xcopy C:\*. * E:\*. * /e /v /s /y
```

Das wäre dann schon praktisch alles. Damit könnte unser BATCH-Programm fertig sein. Ich denke aber, Sie wollen es etwas differenzierter haben. Der RAM-Stick hat keine so große Kapazität, dass das einige Gigabyte große komplette System draufpassen würde. Wir müssen also auswählen, welche Dateien wir sichern wollen. Wie wäre es mit den Eigenen Dateien?

Also programmieren wir als letzte Zeile stattdessen:

```
xcopy C:\EIGENE~1\*. * E:\EIGENE~1\*. * /e /v /s /y
```

Aber Vorsicht:

- 1) Ist bei Ihnen der Pfad zu den Eigenen Dateien so einfach? Dieses Beispiel entstammt einem Win98SE-Rechner. Bei WIN XP kann das schon viel schwieriger sein.
- 2) Achtung bei der Pfadbezeichnung EIGENE~1. Was kommt damit bei Ihrem Rechner heraus? EIGENE BILDER vielleicht? Dann müssen Sie C:\EIGENE~2*. * als Pfadbezeichnung probieren!
- 3) Vorsicht mit schreibgeschützten Dateien! Wenn Sie schreibgeschützte Dateien schon auf dem Stick mittels BACKUP abgespeichert haben, was ja problemlos geht, dann wird es beim nächsten Backup Probleme geben. Denn Sie haben ja jetzt eine schreibgeschützte Datei auf dem RAM-Stick vorliegen, und diese darf nicht überschrieben werden. Also wird DOS eine Fehlermeldung geben und die weitere Bearbeitung des copy-Befehls abbrechen! Wenn Sie hier nicht aufpassen, so klappt die Datensicherung nicht. Wenn Sie aber nicht merken, dass die Datensicherung nicht geklappt hat, dann ist die Datensicherung völlig wertlos. Lesen Sie also die Meldungen auf dem Bildschirm mit, während die Datensicherung läuft. Auch wenn das langweilig ist.

Wenn Sie wollen, können Sie als letztes hinter die Datensicherung noch ein paar Zeilen anhängen, die dem Benutzer sagen, dass die Datensicherung erfolgt ist. Sie benutzen dazu den Befehl echo.

Jetzt viel Erfolg beim Programmieren!

Sobald Sie das BATCH-Programm richtig zum Laufen gebracht haben, können Sie eine Verknüpfung zum Desktop anlegen. Wie dies geht? Ganz einfach: Rechtsklicken Sie auf BACKUP.BAT und klicken Sie in dem Menü auf Verknüpfung. Ziehen Sie die Verknüpfung mit dem Mauszeiger auf den Desktop. Ändern Sie, wenn Sie wollen, den Namen der Verknüpfung durch Rechtsklicken auf das Symbol.

Jetzt haben Sie ein brauchbares Werkzeug zur Datensicherung. Später, wenn Sie den RAM-Stick einmal wieder auslesen, können Sie stets feststellen, wann Sie die Datensicherung durchgeführt haben.

Viel Erfolg!

---=== Ende von BATCHTIPP.TXT ===---