

Arbeiten mit Batch-Dateien

Carl-Theodor Olivet

Einführung

In dieser Abhandlung wird dargestellt, wie man eine sinnvolle Stapelverarbeitungsdatei (Batch-Datei) erstellen kann, um auf einfache Weise wiederholt benötigte Programmaufrufe oder das Kopieren von Dateien, das Formatieren von Datenträgern, die Ansicht eines Verzeichnisbaumes schnell vornehmen zu können.

Eine Batch-Datei ist nichts anderes als eine abgespeicherte Stapelung von Befehlsfolgen, die einzugeben sind, wenn man dem Computer ein bestimmtes Verhalten abverlangt. Man würde die Befehlsfolgen normalerweise einzeln eingeben. Die Batch-Datei macht solche Wiederholungen entbehrlich, sie ist lediglich mit ihrem Namen aufzurufen und arbeitet dann alle in ihr abgelegten Befehle ab. Das ist eine wesentliche Arbeitserleichterung.

Wer die DOSSHELL von MS-DOS 4.01 benutzt, wird festgestellt haben, daß Sie einem die Eingabe einzelner Befehlsfolgen bereits ganz gut abnimmt. Sie ist nur ziemlich überladen, viele Sachen braucht man kaum oder selten, vor allem arbeitet sie recht behäbig und ohne Mausbenutzung auch reichlich umständlich. Und sie beansprucht als speicherresidentes Programm Hauptspeicher, was zu Problemen führen kann (z.B. ist dann bei Anwendung des „Harvard Projekt Manager“ der Hauptspeicherplatz zu klein). Nicht umsonst wird von PC-Anwendern trotz einer vorhandenen DOS-SHELL gerne auf das Programm PC-Tools (von Central Point Software) zurückgegriffen. Aber auch das kann lästig werden, wenn für einfache Zugriffsabsichten häufig nur kleinere Befehlsfolgen erforderlich sind. Für solche einfachen Befehlsfolgen ist eine Batch-Datei gerade richtig.

Ein Beispiel soll das verdeutlichen:

Angenommen Ihr Programm KFZ befindet sich im Subdirectory (Unterverzeichnis) RECHNEN. Der Pfad dorthin lautet zum Beispiel

C:/JURA/GERICHT/RECHNEN KFZ. Haben Sie dieses Programm häufiger in Gebrauch, dann bietet es sich an, eine Batch-Datei mit Namen RECHNEN.BAT auf der Betriebsebene des Computers anzulegen. Um das KFZ-Programm aufzurufen, brauchen Sie dann auf der Betriebsebene nur den Befehl RECHNEN einzugeben; die Befehlsfolge greift sich das System dann aus der Batch-Datei, und das Programm wird allein mit dem eingegebenen Befehl rechnen gestartet.

Dieses Beispiel ist einfach und zeigt nur schwach, wozu eine Batch-Datei fähig ist. Ausgefeilte Batch-Dateien können Großartiges leisten. Es lohnt, sich damit näher zu beschäftigen.

Erster Versuch – COPY CON

Eine Batch-Datei benötigt unbedingt den Anhang (die Extension) „.BAT“, weil das Betriebssystem nach ausführbaren Dateien – in der Reihenfolge – „.COM“ „.EXE“ „.BAT“ sucht. Würden Sie eine Batch-Datei RECHNEN.TXT benennen, würde der Aufruf RECHNEN oder RECHNEN.TXT nur die Fehlermeldung bewirken, eine solche Datei sei nicht vorhanden.

Sie können zur Erstellung einer Batch-Datei mit dem Befehl COPY CON arbeiten, was soviel bedeutet wie „Kopiere von der Konsole/dem Keyboard/der Tastatur aus“ in die Datei. Um bei dem Beispiel zu bleiben, wäre also COPY CON RECHNEN.BAT und abschließend (RETURN) (Eingabe-Taste) einzugeben. Der Cursor (Blinker) springt daraufhin in die nächste Zeile. Hier geben Sie ein: ECHO OFF. Das bedeutet, daß Sie die nachfolgenden Befehlsfolgen beim späterem Aufruf der Batch-Datei nicht auf dem Bildschirm sehen wollen. Diese Zeile schließen Sie mit (RETURN) ab. Der Cursor springt dadurch in die nächste Zeile. Hier geben Sie ein: CALL JURA/GERICHT/RECHNEN

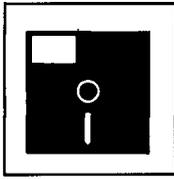
Batch-Dateien erleichtern die Arbeit

DOS-Shell – behäbige Kolosse

Batch-Dateien: Klein, flexibel, leistungsfähig

Dateien erstellen mit COPY CON

Carl-Theodor Olivet ist Richter am LG Lübeck und Autor der Sammeljurium-Reihe, einer Programmsammlung für Juristen (vgl. jur-pc 90, 155) sowie des Buches „Computerprogramme für Juristen“.



Beenden der Eingabe:
Mit <F6> oder <STRG> Z

Auf jedem MS-DOS PC
verfügbar:
Der Zeileneditor EDLIN

Der EDLIN-Prompt:
Das Sternchen „*“

Dateien auf dem Bildschirm lesen:
TYPE

Einfüge-Modus:
Zuschalten mit i,
Verlassen mit <STRG> C

Zeile löschen: d

Datei listen: l

Das Ändern einer Zeile

KFZ<RETURN>. Nun wollen Sie die Eingaben beenden. Dies geschieht mit der Funktionstaste <F6> oder indem Sie bei niedergehaltener <STRG>-Taste ein Z anschlagen (in beiden Fällen erscheint dann die Zeichenfolge/Z auf dem Bildschirm) und die Eingabe mit <RETURN> „abschicken“. Der Computer speichert die Datei dann unter dem Namen RECHNEN.BAT, die Batch-Datei ist fertig.

Zweiter Versuch – der Zeileneditor EDLIN

Wenn eine Batch-Datei umfangreicher ausfallen soll, empfiehlt es sich, sie mit dem von MS-DOS zur Verfügung gestellten Zeileneditor EDLIN zu erstellen. Mit COPY CON ist es nämlich nicht möglich, in die Datei nachträglich Korrekturen einzufügen, ohne die Datei völlig neu zu schreiben. Mit EDLIN erstellt man die Batch-Datei so: Rufen Sie EDLIN mit dem gewünschten Batch-Dateinamen auf: EDLIN RECHNEN.BAT<RETURN>. In der nächsten Zeile erscheint jetzt ein Sternchen (*), das Aufforderungszeichen für Eingaben. Hier sollen nicht alle Funktionen des Editors besprochen werden, wir beschränken uns auf die wichtigsten Elemente. Hinter dem Sternchen geben Sie den Buchstaben i<RETURN> ein (i = insert). Es erscheint eine Zeilennummer (1). Die Befehlsfolgen können nun eingetippt werden. Jede Zeile muß mit <RETURN> abgeschlossen werden. Es beginnt wieder mit ECHO OFF (zur Bedeutung s.o.). In Zeile 2 schreiben Sie CLS (bedeutet clear screen/lösche den Bildschirm). In Zeile 3 tragen Sie die bereits genannte Befehlsfolge ein: CALL JURA/GERICHT/RECHNEN/KFZ. Jeder selbständige Befehl gehört also in eine eigene Zeile! Um die Batch-Datei abzuschließen, halten Sie jetzt <STRG> nieder und tippen C. Der Cursor springt dann in die nächste Zeile, und es erscheint – ausgerückt – ein Sternchen. Damit haben Sie zwar den Einfügemodus, nicht aber schon die Datei verlassen. Um diese abzuschließen und zu speichern, geben Sie den Buchstaben e<RETURN> ein (bedeutet end/Ende). Die Batch-Datei ist damit fertig. Ohne EDLIN benutzen zu müssen, können Sie sich den Inhalt dann mit dem Befehl TYPE RECHNEN.BAT<RETURN> ansehen. Sehen Sie einen Fehler in dem Listing, starten Sie EDLIN RECHNEN.BAT. Es erscheint dann das bekannte Sternchen. Nun tippen Sie l<RETURN> (= list). Die Datei wird dadurch aufgelistet und kann nun bearbeitet werden (*20<RETURN> würde ab Zeile 20 auflisten).

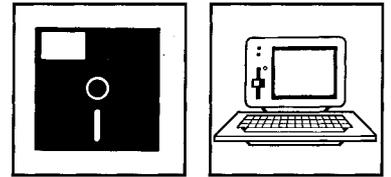
EDLIN-Befehle

Zur Bedienung des Zeileneditors muß man einige Editierbefehle parat haben. Vier kennen wir bereits, i für insert/einfügen, <STRG> C für das Verlassen des Einfügemodus, e für Ende und l für list/auflisten. Wichtig ist noch d für delete/löschen. Mit diesen Befehlen läßt sich schon ganz gut arbeiten (obgleich es noch wesentlich mehr Befehle gibt).

Wollen Sie Zeile 1 löschen, so verlassen Sie – falls der Einfügemodus eingeschaltet ist – zunächst diesen Modus mit <STRG> C. Neben das nun in der nächsten Zeile ausgerückt erscheinende Sternchen schreiben Sie ld<RETURN> (das bedeutet: lösche Zeile 1). Es erscheint erneut ein Sternchen in der folgenden Zeile. Mit l<RETURN> wird die geänderte Batch-Datei nun aufgelistet (l = list/auflisten). Sie sehen, daß der Befehl ECHO OFF gelöscht ist, die Zeilennummer aber noch besteht. Das Löschen bewirkt also eine Entfernung des Zeileninhalts und ein Aufrücken der nachfolgenden Zeilen, Zeilennummer 1 selbst bleibt aber erhalten. Das ist deswegen bedeutsam, weil Sie zum Löschen von Zeile 1-3 nicht ld<RETURN>, 2d<RETURN>, 3d<RETURN> eingeben dürfen. Das löscht Zeilen, die Sie nicht löschen wollen, weil jeder Löschvorgang die übrigen Zeilen nachrücken läßt. Nach dem Entfernen einer Zeile sollten Sie die Datei vorsichtshalber auflisten (*l), um zu sehen, was bereits gelöscht wurde.

Soll ECHO OFF wieder eingefügt werden, dann geben Sie li<RETURN> ein (das bedeutet: füge vor den Inhalt von Zeile 1 ein). In der nächsten Zeile springt das Sternchen auf eine eingerückte Position. Geben Sie jetzt ECHO OFF<RETURN> ein, verlassen Sie den Einfügemodus mit <STRG> C und listen Sie die Datei mit l<RETURN>. Sie sehen dann, daß der Inhalt von Zeile 1 in Zeile 2 gewandert und Zeile 1 neu belegt worden ist.

Wünschen Sie in Zeile 1 statt ECHO OFF ein ECHO ON, dann geben Sie, (nach dem Verlassen des Einfügemodus) neben dem ausgerückte Sternchen die Nummer 1 ein.



Zeile 1 erscheint nun und in der nächsten Zeile ist durch das eingerückte Sternchen der Einfügemodus eingeschaltet. Mit der Funktionstaste (F1) können Sie den Zeileninhalt der Zeile 1 ausrollen. Stoppen Sie nach dem Wort ECHO, geben Sie ON ein und bestätigen Sie mit (RETURN). Das Sternchen springt wieder aus dem Einfügemodus. Mit (RETURN) (auflisten) sehen Sie das Listing nun in korrigierter Form.

Befehle zur Stapelverarbeitung

Es gibt nur wenige Befehle, mit denen man die Stapelverarbeitung steuern kann. Den Befehl ECHO kennen Sie bereits. ECHO ON läßt die Befehlsausgabe auf dem Bildschirm sein, ECHO OFF unterdrückt die Befehlsausgabe auf dem Bildschirm. Der Befehl CALL ruft eine andere ausführbare Datei oder ein ausführbares Programm auf. Zum Starten von Programm mit der Extension EXE oder COM ist CALL nicht unbedingt erforderlich. Anders verhält es sich mit dem Aufruf einer weiteren Batch-Datei. Ohne die Verwendung von CALL wird die erste Batch-Datei durch den Aufruf der zweiten beendet. Die Verwendung von CALL gestattet dagegen die Ausführung der zweiten Stapeldatei, ohne den Lauf der ersten abzubrechen. Der Befehl IF setzt eine Bedingung, der Befehl GOTO setzt einen Sprungverweis zu einer Marke, ab der die Datei weiter abgearbeitet werden soll (Marken werden mit einem vorangestellten Doppelpunkt dargestellt, z.B. :Marke1). PAUSE hält die Abarbeitung der Datei an und setzt sie erst bei Tastendruck wieder fort. Weitere Stapelverarbeitungsbefehle sollen hier außer acht bleiben.

Platzhalter

Wesentlich flexibler wird eine Stapelverarbeitung mit Hilfe von Platzhaltern, die dafür sorgen, daß mit einer Datei verschiedene Dinge ausgeführt werden können. Platzhalter sind Variablen, die in der Datei die Stelle eines Wertes repräsentieren, der beim Aufruf der Datei eingegeben wird. Platzhalter werden als % neben einer Zahl dargestellt. Der geringste ist %0, der höchste normalerweise %9. Benutzen Sie mehr als 10 Platzhalter (z.B. %10), sind Besonderheiten zu beachten, die hier nicht interessieren. Die Variablen werden beim Aufruf der Batch-Datei belegt. Angenommen in unserer KFZ-Befehlszeile steht CALL C:/JURA/GERICHT/RECHNEN/%1 und in dem Directory RECHNEN befinden sich die Programme KFZ und BLUTALKO, dann rufen Sie das Blutalkoholprogramm mit RECHNEN BLUTALKO (RETURN) auf. Das KFZ-Programm würde mit RECHNEN KFZ (RETURN) gestartet.

*Bildschirm-Echo unterdrücken:
ECHO OFF*

Programm-Aufruf mit CALL

*Bedingung überprüfen: IF,
Sprungmarke setzen: GOTO*

Variablen: %0 bis %9

Die Batch-Datei TOTAL.BAT

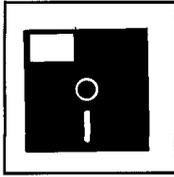
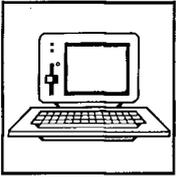
Mit diesen Kenntnissen läßt sich schon eine stattliche Batch-Datei erstellen. Sie soll TOTAL.BAT heißen. Um TOTAL.BAT komfortabel anwenden zu können, erstellen wir zwei Batchdateien. Die eine (nur), um eine benutzerfreundliche Oberfläche zu schaffen, die andere, um die Befehle auszuführen. Die zweite Batch-Datei soll #.BAT heißen. Sie könnte auch ohne TOTAL.BAT aufgerufen werden, wie wir am Beispiel RECHNEN gesehen haben. Die erste Batch-Datei ist nach dem Aufruf von EDLIN TOTAL.BAT (RETURN) schnell erstellt:

```
echo off
CLS
echo      Beachten Sie      : Von Feld zu Feld gehen Sie mit dem Tabulator.
echo                        Die un belegten Felder zwischen zwei belegten
echo                        Feldern müssen mit dem Zeichen - belegt sein.
echo                        „Datei“ muß nur bei der Wahl KOPIEREN mit der
echo                        Extension (.exe / .com usw.) versehen sein.
echo
echo
echo      -----
echo      Ausführen (1) Kopieren (2)  Formatieren (3)  Verzeichnis (4)
echo      -----
echo      #   Nr      LaufWrk Pfad 1   Pfad 2   Datei   Ziel-Laufwerk
echo      -----'-----'-----'-----'-----'-----'-----'-----
```

Eine benutzerfreundliche Oberfläche

Gestartet wird diese Oberfläche mit dem Befehl TOTAL (RETURN). Das #- Zeichen (in der vorletzten Zeile) bedeutet, daß die #.BAT aufgerufen werden soll. Nach Eingabe des

Die Bedienung der Oberfläche



Die Datei #.BAT

gelangen Sie mit der <TAB>-Taste (Tabulator) zu Nr. Hier geben Sie eine der 4 Befehlsziffern ein, z.B. Ziffer 1. Drücken Sie nun wieder <TAB>. Wenn sich das auszuführende Programm – wie diese Batch-Datei – auf der Betriebsebene befindet, brauchen Sie keine Laufwerks- und keine Pfadangabe. Geben Sie dann einen „-“ (bei LaufWrk) ein, tippen <TAB>, erneut einen „-“ bei Pfad 1 und <TAB>, wieder einen Strich bei Pfad 2 und nach einem letzten <TAB> geben Sie den Dateinamen ein (die Angabe der Extension ist nicht erforderlich) und beenden die Eingabe mit <RETURN>. Die Rubrik Ziel-Laufwerk brauchen Sie nicht (wird nur zum Kopieren gebraucht).

Nun wird eine Fehlermeldung ausgegeben, da die #.BAT noch gar nicht existiert, also nicht aufgerufen werden kann. Sie soll jetzt erstellt werden. Nach dem Aufruf von EDLIN #.BAT<RETURN> geben Sie ein:

```

echo off
CLS

if %1 == 3 goto festplattenschutz

if %1 == 4 tree %2:
if %1 == 4 goto ende

if %2 == %3 goto einzel3
if %3 == - goto einzel1
if %4 == - goto einzel2
if %1 == 2 copy/b %2:%3/%4/%5 %6:
GOTO ende

: einzel1
if %1 == 2 copy/b %2:%5 %6:
if %1 == 2 goto ende
call %2:%5
GOTO ende

: einzel2
if %1 == 2 copy/b %2:%3/%5 %6:
if %1 == 2 goto ende
call %2:%3/%5
GOTO ende

: einzel3
if %1 == 2 copy/b %5 %6:
if %1 == 2 goto ende
call %5
GOTO ende

: festplattenschutz
if %2 == c echo Festplatte formatieren ! Abbruch durch Strg+C
if %2 == c pause
format %2:

: ende

```

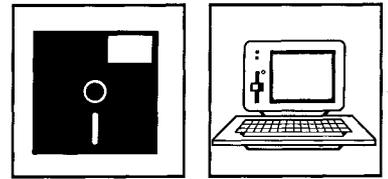
Die Platzhalter sind:

- %1 für Nr
- %2 für LaufWrk
- %3 für Pfad 1
- %4 für Pfad 2
- %5 für Datei
- %6 für Ziel-Laufwerk

Man sieht (als selbstverständlich), daß eine Batch-Datei alle DOS-Befehle verarbeitet (hier copy, format, tree). Das „==“-Zeichen ist ein auf Übereinstimmung vergleichender Relationsoperator.

Die Stapelverarbeitung beginnt mit einem Ausschalten der Bildschirmanzeige (ECHO

Vergleichen mit „==“



OFF) und klärt den Bildschirm (CLS). Nun wird geprüft, ob unter Nr die 3 (formatieren) eingegeben wurde. Falls ja, verzweigt das Programm zur Marke FESTPLATTENSCHUTZ, da nicht versehentlich die Festplatte formatiert werden soll. Wurde unter LaufWrk tatsächlich ein C eingegeben, erfolgt mit ECHO (zeige auf dem Bildschirm) eine Abfrage (Festplatte formatieren ?!) und der Befehl PAUSE stoppt die weitere Stapelverarbeitung. Mit <RETURN> wird der Ausführung fortgesetzt und durch FORMAT %2: der Datenträger des bezeichneten Laufwerks formatiert.

Haben Sie unter Nr eine 4 (Verzeichnis) eingegeben, dann erscheint auf dem Bildschirm durch den Stapelbefehl IF %1 == 4 TREE ein Verzeichnisbaum der Directories. Der vierte Absatz der Datei (beginnend mit IF %2 == %3 GOTO einzel3) prüft, ob ein Pfad gesetzt wurde. Da %2 und %3 nach den Konventionen in TOTAL.BAT nur dann gleich sein können, wenn beide mit einem „-“ belegt sind, muß nach den allgemeinen Pfadregeln auch Pfad2 mit einem „-“ belegt worden sein. Die Datei schließt aus %2 == %3 also folgende Eingabe

```
#      Nr      LaufWrk Pfad 1   Pfad 2   Datei   Ziel-Laufwrk
-----
#      1 oder 2  -         -         -         Name
```

und springt zur Marke :einzel3 . Dort wird geprüft, ob unter Nr KOPIEREN (= 2) gewählt wurde. Falls das nicht geschah, kann nur AUSFÜHREN gewählt worden sein, so daß der Befehl CALL %5 abgearbeitet, also das unter DATEI genannte Programm ausgeführt. Prinzipiell gleich arbeiten die beiden der Zeile IF % 2 == %3 ... nachfolgenden Zeilen. Wenn %3 == - ist, also die erste Pfadangabe unterbleibt, dann muß Pfad2 ebenfalls unbelegt sein. Es erfolgt also ein Sprung zu Marke :einzel1 Das soll genügen, Ihnen ein eigenes Erschließen der übrigen Befehle zu ermöglichen. Der Befehl COPY/B bedeutet „kopiere binär“. Sie können stattdessen auch COPY verwenden.

Anhang

Sollten Sie nach häufigem Gebrauch der TOTAL.BAT die dortige Eingabestruktur im Kopf haben, brauchen Sie diese Batch-Datei nicht mehr aufzurufen. Sie können dann die *.BAT sogleich ansprechen, z.B. durch: # 2 a spiele - *.* b(RETURN). Wissen Sie was das bedeutet? Die Batch-Datei *.BAT soll von der Diskette in Laufwerk A und dort aus dem Directory SPIELE alle Dateien auf die Diskette in Laufwerk B kopieren. Häufig wiederkehrende Befehlsfolgen – wie die eingangs zum KFZ-Programm – würde man über dieses System nicht mit den einzelne Pfadnamen aufrufen wollen. Hier bietet es sich an, auf die komplette Befehlsfolge in der TOTAL.BAT mit einer Kennziffer, etwa Kfz = 5 Bezug zu nehmen und in die *.BAT gleich in Zeile 1 die Befehlsfolge IF %1 == 7 CALL JURA/GERICHT/RECHNEN/KFZ und in Zeile 2 IF %1 == 7 GOTO ENDE zu setzen.

*Die Stapelverarbeitung anhalten:
PAUSE*

Die Auswertung von %2 und %3

*Anregungen für eigene
Modifizierungen*