

Literarische und Dokumentarische Datenverarbeitung

Neues zum TUSTEP-Kommando UMWANDLE

(Übertragen von TUSTEP-Dateien zwischen Rechnern, Teil II)

Die Leistungen des Kommandos UMWANDLE umfassen vier Anwendungsbereiche:

- 1) Austausch von TUSTEP-Daten zwischen verschiedenen Rechnern
- 2) Verschlüsselung der Daten (Datenschutz)

3) Archivierung der Daten in komprimierter Form

4) Übergabe und Übernahme von Daten an und von anderen Systemen.

In allen diesen Bereichen gibt es neue Leistungen.

1) Austausch von TUSTEP-Daten zwischen verschiedenen Rechnern

Der direkte Austausch von TUSTEP-Dateien über ein Netz (z. B. ETHERNET mittels FTP) ist nur zwischen Rechnern gleichen Typs (gleiches Betriebssystem und gleiche Rechnerarchitektur) möglich, z. B. zwischen zwei MS-DOS PCs oder zwischen zwei UNIX-Workstations des gleichen Herstellers. In allen anderen Fällen muß für die Übertragung der Daten ein Datenaustauschformat verwendet werden, das die Unterschiede der Rechner bezüglich der Codierung der Zeichen und der Dateistruktur ausgleicht. Dies wurde in BI 91/5+6, S. 12f ausführlich beschrieben.

Für das Umwandeln von Daten ins bzw. aus dem Austauschformat wurden zur Spezifikation MODUS die Angaben TX bzw. XT (ersetzt: -1;x bzw. x;+1) und BX bzw. XB (für Binärdaten) neu eingeführt.

a) Übertragen von Textdaten

Sollen Textdaten im Austauschformat transportiert werden, so sind folgende Angaben zu MODUS zu verwenden:

MODUS=TX (Tustep nach eXport). Die Daten werden im Datenaustauschformat in die ZIEL-Datei geschrieben und dabei wie bei MODUS=-1 umgewandelt (entspricht dem bisherigen Modus -1;x).

MODUS=XT (eXport nach Tustep). Die Daten stehen im Datenaustauschformat in der QUELL-Datei; sie werden ins TUSTEP-Format zurückverwandelt und dabei wie bei MODUS=+1 umgewandelt (entspricht dem bisherigen Modus x;+1).

Beispiel:

```
#umseqfile, sdffile, mo=tx, lo=+
```

Der Inhalt einer TUSTEP-Datei *seqfile* wird im Datenaustauschformat in eine SYSTEM-Datei *sdffile* geschrieben. Die Datei *sdffile* kann dann übers Netz auf andere Rechner übertragen werden, gleichgültig um welche Rechnertypen es sich handelt.

```
#um, sdffile, seqfile, mo=xt, lo=+
```

Die Daten, die im Austauschformat in der SYSTEM-Datei *sdffile* stehen, werden im TUSTEP-Format in die Datei *seqfile* geschrieben.

b) Neu: Übertragen von Binärdaten

Binärdaten, d. h. Daten in denen alle 256 in einem byte möglichen Zahlenwerte vorkommen können, entstehen in TUSTEP bei der AUSGABE-Datei des Satz-Programms oder beim Verschlüsseln von Daten (siehe unten). Um diese Daten übers Netz zwischen verschiedenen Rechnertypen zu verschicken, gibt es ein eigenes Datenaustauschformat für Binärdaten:

MODUS=BX (Binär nach eXport). Die Daten werden im Datenaustauschformat in die ZIEL-Datei geschrieben. Jedes Zeichen (byte) wird dabei in zwei Hexadezimalziffern umgewandelt. Damit können binäre Daten in ein Export-Format gebracht werden.

MODUS=XB (eXport nach Binär). Die Daten stehen im Datenaustauschformat als Folge von Hexadezimalziffern. Sie werden ins TUSTEP-Format zurückverwandelt; die Hexadezimalziffern werden dabei paarweise in Zeichen (bytes) umgewandelt.

2) Datenverschlüsselung

Eine weitere Leistung des Kommandos `UMWANDLE` ist das Verschlüsseln von Daten, um zum Datenschutz die Daten für andere unlesbar zu machen. Neu ist, daß durch den `MODUS=BX` die verschlüsselten Daten in ein Austauschformat gebracht werden können, um die Daten übers Netz zu schicken. So bleibt auch im Netz der Datenschutz gewahrt.

Dringend zu empfehlen ist das Verschlüsseln der Daten beim Filetransfer via *anonymous* auf dem `MAILSERV`. Doch sollte man bei sensiblen Daten auch für den direkten Filetransfer diese Verschlüsselung wählen, um ein Mitlesen während des Transports zu vereiteln.

Beispiel:

Filetransfer via *anonymous* auf dem `MAILSERV`

Der direkte Filetransfer mit FTP – neben dem Verpacken von (kurzen) Dateien als `MAIL` zur Zeit die einzige Möglichkeit des Datentransfers – hat den Nachteil, daß das Paßwort der Login-ID des Austauschpartners bekannt sein muß, um dorthin eine Datei zu senden oder um von dort eine Datei abzuholen. Will man die Bekanntgabe des Paßworts und die damit verbundenen Gefahren (wie Löschen, Überschreiben und Kopieren von Dateien) vermeiden, bietet es sich an, den Filetransfer über *anonymous* auf dem `MAILSERV` abzuwickeln (vgl. dazu BI 91/3+4, S. 2).

Will Benutzer/in A zu Benutzer/in B (beide befinden sich auf einem Rechner am `ETHERNET`, gleichgültig ob im `ZDV` oder in einer anderen Stadt) eine Datei schicken, so hat das Verfahren folgenden Ablauf:

- A schickt die Datei an *anonymous* auf `MAILSERV` (a) und b)).
- B holt die Datei von *anonymous* auf `MAILSERV` (b) und c))

Aus Gründen des Datenschutzes (solange die Datei unter *anonymous* auf dem `MAILSERV` liegt, kann sie von der ganzen Welt gelesen, abgeholt, überschrieben oder gelöscht werden) empfiehlt es sich, für die Übertragung die Daten zu verschlüsseln, so daß wenigstens niemand die Daten interpretieren kann.

a) Verschlüsseln der Daten und Umwandeln ins Datenaustauschformat

Vor dem Filetransfer werden von Benutzer/in A die Daten einer `TUSTEP`-Datei im Austauschformat in eine `SDF`-Datei geschrieben und gleichzeitig verschlüsselt:

```
#umseqfile, sdffile, m=bx, l=+, s=*  
Und Simson liebte eine Frau im Tal  
Sorek. Ihr Name war Delila.  
*eof
```

Der Schlüssel ist ein beliebiger Text, der mindestens 40, maximal 400 Zeichen lang ist. Groß- und Kleinschreibung, Satzzeichen und Leerzeichen sind von Bedeutung.

b) Transfer der Datei *sdffile* zur Login-ID *anonymous* auf den `MAILSERV` des `ZDV`

Die Übertragung der Datei erfolgt über `ETHERNET` mittels `FTP` mit folgenden Befehlen:

```
% ftp mailserv  
Name (mailserv:userid_a): anonymous  
Password: beliebig  
ftp> cd scratch  
ftp> dir
```

Mit `dir` bekommt man die Liste der schon vorhandenen Dateien und kann bei der Übertragung beim folgenden Befehl ggf. einen neuen Namen vergeben, um vorhandene Dateien nicht zu überschreiben:

```
ftp> put sdffile  
ggf.: ftp> put sdffile newname  
ftp> quit
```

Die Datei kann nun von Benutzer/in B (genauer gesagt von der ganzen Welt) vom `MAILSERV` abgeholt werden.

c) Holen der Datei von *anonymous*

```
% ftp mailserv  
Name (mailserv:userid_b): anonymous  
Password: beliebig  
ftp> cd scratch  
ftp> get sdffile
```

Vorsicht! Eine gleichnamige Datei auf dem eigenen Rechner wird ohne Warnung überschrieben; ggf. kann bei der Übertragung ein neuer Name vergeben werden:

```
ggf.: ftp> get sdffile newname  
ftp> delete sdffile
```

Die abgeholte Datei wird mit `delete` auf dem `MAILSERV` gelöscht.

```
ftp> quit
```

Besonderheiten:

Der vollständige Name des `MAILSERV` *mailserv.zdv.uni-tuebingen.de* muß verwendet werden, um ihn von außerhalb des Tübinger Netzes anzusprechen.

Die `COMPAREX` meldet sich nach dem Einloggen mit `FTP` mit dem Prompt *Command:*.

Auf dem PC ist vor dem Befehl ftp noch die Software tcp aufzurufen. Der Name der Datei auf dem Zielrechner wird vor der Übertragung erfragt.

d) Entschlüsseln und Rückwandeln der Daten

Die Daten müssen nach der Übertragung aus dem Datenaustauschformat in eine TUSTEP-

Datei umgewandelt und dabei entschlüsselt werden. Dazu muß exakt der gleiche Schlüssel angegeben werden, der beim Verschlüsseln verwendet wurde:

```
#umsdfile, seqfile, m=xb, l=+, s=*
```

```
Und Simson liebte eine Frau im Tal  
Sorek. Ihr Name war Delila.
```

```
*eof
```

3) Archivierung von Daten in komprimierter Form.

Ein neuer Modus im Kommando UMWANDLE erlaubt das Komprimieren von Daten auf ca. 50–60% ihres ursprünglichen Platzes.

Die komprimierten Dateien eignen sich nicht zum Austausch übers Netz oder zur Weiterverarbeitung, sondern dienen ausschließlich der Archivierung. Sie sind erst wieder verwendbar, wenn sie in TUSTEP dekomprimiert werden.

MODUS=TK (Tustep nach Komprimiert).

Die Daten werden in komprimierter Form in die ZIEL-Datei (SDF- oder SEQ-Datei) geschrieben.

MODUS=KT (Komprimiert nach Tustep).

Die Daten stehen in komprimierter Form in der QUELL-Datei: sie werden dekomprimiert und in die ZIEL-Datei (SEQ-Datei) geschrieben.

4) Übergabe und Übernahme von Daten an und von anderen Systemen

Das Kommando UMWANDLE ist auch die Schnittstelle zu anderen Systemen (System-Daten, andere Textverarbeitungssysteme, Datenbanken, etc.). Für diese Funktion wurde die Spezifikation CODE erweitert und das Kommando um eine neue Spezifikation NL ergänzt. Für in TUSTEP nicht definierte Zeichen wurde eine Ersatzdarstellung eingeführt.

a) Spezifikation CODE: neue Angabe

Mit der Spezifikation CODE wird definiert, wie die Umcodierung der einzelnen Zeichen erfolgen soll. Dabei kann man pauschal Umcodierungen angeben (TUSTEP, SYSTEM, ASCII, EBCDIC) und/oder selbst Codepaare aufzählen, die zur Umwandlung herangezogen werden sollen.

Neu ist die Möglichkeit, die Spezifikationswerte TUSTEP oder SYSTEM mit der Angabe von hexadezimalen Code-Paaren zu kombinieren. Dies erleichtert z. B. bei der Übernahme von MS-DOS-Daten die Umwandlung von Codes, für die in den entsprechenden Tabellen kein TUSTEP-Zeichen vorgesehen ist.

```
CODE=TUSTEP'xx:yy'zz:qq'...
```

Die Daten sollen wie für die Sichtgeräte-Eingabe entsprechend dem mit DEFINIERE eingestellten Code umcodiert werden. Zusätzlich soll das Zeichen mit dem hexadezimalen Code xx in das Zeichen mit dem

hexadezimalen Code yy, das Zeichen zz in qq, etc. umcodiert werden.

Der Spezifikationswert TUSTEP muß dabei immer an erster Stelle angegeben werden.

```
CODE=SYSTEM'xx:yy'zz:qq'...
```

Die Daten sollen wie für die Sichtgeräte-Ausgabe entsprechend dem mit DEFINIERE eingestellten Code umcodiert werden. Zusätzlich soll das Zeichen mit dem hexadezimalen Code yy, das Zeichen zz in qq, etc. umcodiert werden.

Der Spezifikationswert SYSTEM muß dabei immer an erster Stelle angegeben werden.

b) Ersatz für nicht darstellbare Codes

Neu ist die Ersatzdarstellung für Zeichen, denen kein TUSTEP-Zeichen entspricht.

Beim Umwandeln von einer SDF-Datei in eine SEQ-Datei werden Zeichen der QUELL-Datei (SDF-Datei), die in TUSTEP nicht darstellbar sind, in die ihrem Code entsprechenden Hexadezimalziffern umgewandelt und in der Form #[xx] in die ZIEL-Datei geschrieben. Diese Zeichenfolgen können dann mit den üblichen TUSTEP-Mitteln (Editor, KOPIERE) gesucht und – auch in Kombination mit umgebenden Zeichen – ausgetauscht werden. Damit ist es möglich, beliebige Daten nach TUSTEP zu übernehmen.

Beim Umwandeln einer SEQ-Datei in eine SDF-Datei werden in der QUELL-Datei (SEQ-Datei) enthaltene Zeichenfolgen der Form #[xx] in die ZIEL-Datei als das Zeichen mit dem entsprechenden hexadezimalen Code xx geschrieben. Damit ist es möglich, in TUSTEP nicht darstellbare Zeichen in eine SDF-Datei zu schreiben, sofern das System, in das die Daten übergeben werden, dies erfordert.

c) Neue Spezifikation NL (New Line)

Unter den Betriebssystemen UNIX und MS-DOS sind Dateien nicht in Records eingeteilt, sondern in Form von Endlos-Strings (stream) abgelegt. Enthalten solche Dateien nicht den üblichen Code für Zeilenwechsel (0A in UNIX, 0D 0A in MS-DOS), so konnten sie mit UMWANDLE bisher nicht bearbeitet werden, da UMWANDLE spätestens nach 32000 Zeichen einen Zeilenwechsel erwartet. Mit der neuen Spezifikation NL ist es möglich, einen Code anzugeben, der als Zeilentrenner aufgefaßt werden soll.

NL=xx Angabe genau eines hexadezimalen Codes xx.

Wirkung von NL beim Umwandeln einer SDF-Datei in eine SEQ-Datei:

In der ZIEL-Datei wird an den Stellen eine neue Zeile (neuer TUSTEP-Satz) begonnen, an denen das Zeichen mit dem hexadezimalen Code xx in der QUELL-Datei gefunden wird. Das Zeichen selbst wird nicht in die ZIEL-Datei ausgegeben.

Wirkung von NL beim Umwandeln einer SEQ-Datei in eine SDF-Datei:

In der ZIEL-Datei wird an den Stellen das Zeichen mit dem hexadezimalen Code xx eingefügt, an denen in der QUELL-Datei eine Zeile (TUSTEP-Satz) endet.

Damit ist es möglich, Endlos-Strings (stream) abweichend vom voreingestellten Zeilentrenner (0A in UNIX, 0D 0A in MS-DOS) zu erhalten.