

Split_D

COLLABORATORS

	TITLE : Split_D		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		November 24, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Split_D	1
1.1	Split Guide	1
1.2	Copyright	1
1.3	Einführung	2
1.4	Befehlsbeschreibung	2
1.5	Technische Details	4
1.6	Das Programm und sein Autor	4
1.7	Geschichte des Programms	4

Chapter 1

Split_D

1.1 Split Guide

Split - Version 1.0
Copyright © 1994 Marius Chwalczyk
Alle Rechte Vorbehalten

Copyright
Einführung
Befehlsbeschreibung
Programm und Autor
Geschichte

1.2 Copyright

Das Paket "Split - Version 1.0" steht unter Copyright © 1994 von Marius Chwalczyk. Alle Rechte sind vorbehalten.

Es ist VERBOTEN das Paket zu verkaufen oder sonstwie weiterzugeben. Das Paket kann nur unter folgenden Bedingungen frei weitergegeben werden:

1. nur minimale Gebühr (Kopierkosten, Datenträgerpreis) sind bei der Weitergabe erlaubt.
2. alle hier aufgeführten Dateien müssen in ihrem originellen Zustand, ohne irgendwelche Änderungen enthalten sein:

Split	3292 - der Befehl
Copyright-LiesMich	1442 - diese Datei
Copyright-ReadMe	1192 - diese Datei in englisch
Split_D.guide	8756 - deutsche Dokumentation
Split.guide	7354 - englische Dokumentation
Split.info	843 - \
Copyright-LiesMich.info	835 -
Copyright-ReadMe.info	835 - > Piktogramm-Dateien

```
Split_D.guide.info      841 - |  
Split.guide.info        841 - /
```

3. die Dateien können gepackt werden (z. B. mit LhA oder Lharc).
Es ist aber kein Crunchen (das Packen mit beigehaltener Lauffähigkeit) der ausführbaren Dateien erlaubt.

Falls das Paket in PublicDomain-Serien aufgenommen wird, bitte ich (der Autor) um eine Information darüber. Für eine beigelegte, kostenfreie Kopie der Ausgabe danke ich voraus!

Das Programm ist "wie es ist", jede Benutzung ist auf eigene Risiko.

1.3 Einführung

Split ist ein CLI-Befehl. Es läuft nur ab der Release 2 (Version 37) von Amiga-Betriebssystem.

Man kann damit zum Beispiel beliebige Dateien in kleinere teilen, wenn sie zu groß sind, um sie auf eine Diskette zu kopieren. Eine so geteilte Datei kann später mit dem System-Befehl "Join" zusammengesetzt werden. Da die Teildateien numeriert werden, ist es einfach, sie in der richtigen Reihenfolge anzugeben.

Die Aufteilung kann auf vier verschiedene Arten erfolgen und wird mit den Optionen bestimmt.

1.4 Befehlsbeschreibung

SPLIT

Format: SPLIT <Name> [TO <Name>] [<Größe>] [PARTS <Anzahl>]
 [LINES <Anzahl>] [FMTCMD <Zeichen>] [QUIET] [ABOUT]

Schablone: FROM/A,TO/K,SIZE/N,PARTS/K/N,LINES/K/N,FMTCMD=%/K,QUIET/S,ABOUT/S

Funktion: Teilen von Dateien.

Beschreibung:

Split kopiert eine gegebene Datei <Name> in mehrere, numerierte Teildateien (auch nur eine ist möglich). Das kopieren kann jederzeit mit der Tastenkombination "Ctrl-C" abgebrochen werden. Unter Umständen kann der Befehl aber erst nach einer fertigen Teildatei aufhören zu kopieren.

Normalerweise werden die Teildateien in das aktuelle Verzeichnis mit einem gleichen Basisnamen und einer Endung ".%" kopiert, wobei "%" die laufende Nummer der Teildatei darstellt. Also die erste Teildatei wird den Namen "<Name>.1" tragen. Mit der Option TO <Name> kann ein anderes Verzeichnis und/oder ein anderer Basisname bestimmt werden.

Wenn der Basisname das vorgegebene Format-Zeichen '%', oder das mit der Option

FMTCMD <Zeichen> bestimmtes, enthält, wird die laufende Nummer an der Stelle des Format-Zeichens eingesetzt. Es wird nur das erste Zeichen ersetzt. Falls es aber nicht ersetzt sein soll, muß ein anderes gegeben werden.

In jedem Fall achtet der Befehl, daß keine gleichen oder zu langen (maximal 30 Zeichen) Namen entstehen. Es werden aber die bereits mit solchen Namen existierende Dateien ohne Anfrage überschrieben.

Das Argument <Größe> muß eine positive Zahl sein und gibt die Größe in Bytes der zu erstellten Teildateien an.

Mit der Option PARTS <Anzahl> kann bestimmt werden, in wieviele Teilen die Datei geteilt sein soll. Die Größe der Teilen wird dabei berechnet. Wenn die Anzahl gleich 1 ist, funktioniert der Befehl wie "Copy" mit Dateien.

Es können auch die beiden vorherigen Argumente zusammen angegeben werden. In diesem Fall werden nicht mehr Teildateien als die gegebene Anzahl erzeugt. Da die Größe auch gegeben ist, kann die letzte Teildatei größer als die gegebene Größe sein. Es ist nützlich, wenn sie z. B. Teil einer 600KB Datei auf eine Diskette mit nur noch 200KB freiem Platz und den Rest auf eine andere kopieren wollen. Es genügt also nur zwei Teile zu erzeugen.

Es kann weniger Teildateien geben als gefordert, falls die Datei zu klein ist.

Die Option LINES <Anzahl> darf nur mit Textdateien benutzt werden. Sie kann auch nicht mit den beiden vorherigen kombiniert werden. Sie ist gedacht, um Textdateien nach der Anzahl der Zeilen teilen zu können. Die Teildateien werden also nicht mehr Zeilen als die gegebene Anzahl enthalten und können alle unterschiedliche Größen haben. Es ist sehr wichtig, daß diese Option nur mit Textdateien benutzt wird. Insbesondere darf die Datei kein NIL-Zeichen ('0x0') enthalten (die Binärdateien haben es meistens), da sonst Fragmente der Datei beim kopieren verloren gehen werden. Der Befehl weigert sich solche Dateien anzunehmen, aber nicht bei jeder. Programmierer bitte die technischen Details lesen.

Die Option QUIET unterdrückt Nachrichten über erstellten Teildateien. Und die Option ABOUT zeigt den Namen des Autors (den meinen) und die Copyright-Anmerkung an.

Beispiele:

```
SPLIT Arc.lha TO ram: 87000 PARTS 3
  Kopiert "Arc.lha" ins Ram-Disk in Teilen zu 87000 Bytes mit den Namen:
  "Arc.lha.1", "Arc.lha.2" und den Rest als "Arc.lha.3".
```

```
SPLIT Devs:Kickstart TO T:50%_Kick_Teil_#_von_2 PARTS 2 FMTCMD=#
  Macht zwei (fast) gleich großen Dateien: "50%_Kick_Teil_1_von_2" und
  "50%_Kick_Teil_2_von_2" im Verzeichnis "T:".
```

```
SPLIT Split_D.guide LINES 100
  Diese Textdatei (mit weniger als 200 Zeilen) wird in zwei kleinere
  Dateien gespaltet.
```

Hinweis: Split ist "pure", d.h. es kann resident gemacht werden.

1.5 Technische Details

Das Problem liegt in der Implementierung der Funktion FGets() aus dos.library. Sie ist gedacht um Zeilen zu lesen. Sie gibt aber leider nicht zurück wieviele Zeichen von der Zeile gelesen wurden. Was nützt mir ein Pointer auf einen Buffer, den ich selbst der Funktion übergebe, als Resultat? Es wäre besser ein Pointer auf das zuletzt gelesene Zeichen oder die Anzahl der Zeichen zu erhalten. Die Kontrolle auf Erfolg wäre mit NULL (bzw. 0) als Resultat auch gegeben. Also wenn ich jetzt eine Zeile mit einem NIL-Zeichen darin lese (ich weiß, sollte nicht sein aber immer hin kann es), dann kann ich unmöglich mehr feststellen, wieviel gelesen wurde.

Und außerdem ist die Funktion FEHLERHAFT. Festgestellt habe ich es mit der Version 37.44. Es wird gesagt, daß die Funktion: FGets(fh, buf, len) nicht mehr als 'len' Zeichen in den Buffer schreibt.

RKRM 3rd Edition, Autodocs:

"... UP TO the number of len specified bytes minus 1 will be copied into the buffer. Hence if a length of 50 is passed and the input line is longer than 49 bytes, it will return 49 characters. (...) The string read in IS null-terminated."

Schöne gesagt, es werden aber 'len' Bytes in den Buffer kopiert und das NIL-Zeichen an der Stelle 'len + 1' gesetzt (hinter dem Buffer-Ende). Und schon beschwert sich MungWall oder es kommt der alte Freund Guru zu besuch. Der Bug ist vielleicht nicht aufgefallen, weil man meistens einen größeren Buffer nimmt und seltener längere Zeile schreibt. Aber immerhin, er ist da. Oder habe ich da etwas falsch verstanden?

Die Lösung: immer 'len - 1' der Länge des Buffers der Funktion übergeben.

1.6 Das Programm und sein Autor

Split wurde in C geschrieben und mit "SAS/C 6.51" auf einem Amiga 3000 mit KS 37.175 und WB 38.35 kompiliert und getestet.

Falls Sie Kommentare, Ideen, Fehlermeldungen haben, oder meine Arbeit belohnen wollen, dann schreiben Sie mir (BITTE die Versionsnummer des Programms angeben) oder rufen Sie mich an.

Ich bin erreichbar per

Post: Marius Chwalczyk
Ziegelstr. 59/165
D-67655 Kaiserslautern
Deutschland

Telefon: 0631-10944

1.7 Geschichte des Programms

1.0 (10.10.93) erste Freigabe.
