

# CP/M 2.2

Ein Kurze Einführung/Gedächtnis-Stütze (von [www.drcrazy.de/nkc/html/cpm.htm](http://www.drcrazy.de/nkc/html/cpm.htm))

CP/M ist eine Reihe von Betriebssystemen, die ab etwa 1974 von Digital Research Inc. unter Gary Kildall entwickelt wurden. CP/M war (neben dem für leistungsstärkere Rechner konzipierten Unix) das erste plattformunabhängige Betriebssystem.

## Allgemeines

### System-Start

Das Betriebssystem CP/M-80 (Control Program for Microprocessors) ist ein diskettenorientiertes System und aus diesem Grund stets nur mit Disketten bzw. Festplatten zu betreiben. CP/M ist von der Systemdiskette in den Arbeitsspeicher zu bringen (booten oder auch Kaltstart). Das erfolgreiche Booten wird durch eine Bildschirmmeldung und dem Erscheinen des System-Prompts ( A> ) angezeigt.

### Prompt

Bereitschaftsmeldung ( A> ) Das A bezeichnet das angemeldete Laufwerk und das «größer als» die eigentliche Bereitschaft zur Annahme von Eingaben über die Tastatur.

### Eingaben

Zulässige Eingaben sind: Kommandos und Programmaufrufe. Alle Eingaben müssen regelmäßig mit einem Wagenrücklauf (er o. carriage return) abgeschlossen werden.

### Laufwerk

Das angemeldete Laufwerk ist das vor dem Prompt bezeichnete. Auf dieses Laufwerk wird nach einem Programmaufruf zugegriffen. Das Laufwerk kann durch Eingabe eines neuen Laufwerknamens und einem Doppelpunkt gewechselt werden. (B: «er») Der Name eines Laufwerkes besteht aus einem Buchstaben (A:, B:, ....P:) und einem Doppelpunkt.

### Dateien

Die auf einer Diskette aufgezeichneten Informationen (Daten) werden nach ihrer logischen Zusammengehörigkeit zu selbständigen Einheiten, sogenannten Dateien (Daten-Kartei) zusammengefaßt. Diese Dateien (Files) können einzeln angesprochen oder benutzt werden.

### Dateinamen

Alle Dateien werden mit einem Namen versehen. Das Ansprechen einer Datei erfolgt stets über den Dateinamen, der aus dem eigentlichen Namen und einer Klassenbezeichnung (extension) besteht. Die Länge des Namens ist auf 8 Zeichen für den Namen und 3 für die Extension begrenzt. Die übliche Schreibweise ist «name.ext»

### Mehrfachnamen

Zu bestimmten Befehlen und Programmen können direkt bei der Eingabe Dateinamen mit angegeben werden. Unter CP/M besteht die Möglichkeit, Mehrfachnamen anzugeben. Diese Namen werden aus den sogenannten «Wildcards» «\*» und «?» zusammengesetzt.  
\*.\* spricht alle auf einer Diskette vorhandenen Dateien an (auch ????????.???)  
.C?? zeigt auf alle Dateien, bei denen der erste Buchstabe der Klassenbezeichnung (Extension) ein «C» ist.

### Dateiinhalte

Dateien können unterschiedliche Inhalte aufweisen.

### **Programmdateien**

Dateien mit der Klassenbezeichnung «COM» enthalten lauffähige Programme, die in binärer Form gespeichert sind.

### **Datendateien**

Textdateien: Im ASCII-Format angelegte Dateien, die direkt auf dem Bildschirm sichtbar gemacht werden können.  
sonstige Datendateien, deren Inhalt nur von speziellen Programmen ausgewertet wird.

### **Programmaufrufe**

Neben den sechs unten genannten residenten CP/M Kommandos werden alle weiteren Befehle mittels spezieller Programme ausgeführt. Ein Programm wird unter CP/M durch Eingabe des Dateinamens ohne Klassenbezeichnung mit abschließendem «return» gestartet.

«Programmname» «cr»  
z.B.: WS «cr» (Starten von WordStar)

### **Diskettenwechsel Warmstart**

Nach jedem Diskettenwechsel ist ein Warmstart durch Eingabe von Control C durchzuführen. Durch das Wechseln einer Diskette wird automatisch ein Schreibschutz eingeschaltet, so daß ein Versuch auf die Diskette zu schreiben, zu einer Fehlermeldung (u. evtl. Datenverlust) führt.

## **residente Kommandos**

### **DIR**

Inhaltsverzeichnis einer Diskette auf dem Bildschirm ausgeben, (directory). Alle auf der Diskette vorhandenen Dateien werden mit ihren vollständigen Namen (name.ext) aufgelistet.

DIR B:\*.COM ergibt ein Verzeichnis aller Dateien mit der Klassenbezeichnung «COM» auf der Diskette in Laufwerk «B:».

### **ERA**

«Name.ext»

Löscht die angegebene Datei. (Erase = Löschen)

Die Verwendung von Mehrfachnamen ist zulässig.

ERA B:.\* löscht alle auf der Diskette im Laufwerk «B:». Sollen alle Dateien gelöscht werden, erscheint eine Sicherheitsabfrage, um versehentliches Löschen zu vermeiden. Erst nach Beantwortung mit «Y» (Yes) wird der Löschkvorgang ausgeführt.

### **REN**

«neuer Name.ext»=«alter Name.ext»

Umbenennen einer Datei. Die als «alter Name.ext» angegebene Datei erhält den unter «neuer Name.ext» bezeichneten Namen; keine Mehrfachnamen.

REN B:NEUNAME.TXT=B:ALTNAMEN.TXT

### **TYPE**

«name.ext» Listet den Inhalt einer Textdatei auf dem Bildschirm. Die Datei darf nur darstellbare ASCII-Zeichen enthalten. Wird versucht, eine Programmdatei «.COM» zu listen, erscheint nur ein großes Durcheinander auf dem Bildschirm. Die Folge könnte ein

undurchschaubarer Zustand des Rechner-Systems sein, der nur durch einen Neu-Start (booten, Kaltstart) zu beheben ist.

## **SAVE**

«Anzahl» «Name.ext» speichert den Inhalt des Arbeitsspeichers in einer Datei ab. Es wird der Inhalt ab Speicherstelle 100H in 256 Byte Blöcken in die Datei «name.ext» heruntergeladen. Die Anzahl der Blöcke ist stets (dezimal) mit anzugeben.

## **USER**

«nummer» schaltet auf den in „Nummer“ gewählten Benutzerbereich auf der Diskette (Festplatte) um. Die Anzahl der zur Verfügung stehenden Bereiche ist 32 (0..31) auf einer Diskette innerhalb eines logischen Laufwerkes. Durch diese Einrichtung werden voneinander unabhängige Arbeitsbereiche auf dem Speichermedium geschaffen, die von anderen USER-Bereichen aus nicht direkt erreicht werden können. (Ausnahme siehe „PIP - kopieren“). Die Einteilung eines logischen Laufwerkes in getrennte Benutzerbereiche ist vor allem zur Kompatibilität mit dem Mehrplatz-BS MP/M gedacht.

# **STAT**

(Status, Statistics) ist ein externer CP/M Befehl. Er wird wie jedes andere Programm aus einer Programmdatei (Klassenbezeichnung «COM») heraus ausgeführt (Eintrag im Inhaltsverzeichnis («DIR») = STAT. COM)

## **Funktion**

Der Befehl (Programm) STAT dient zur Überprüfung oder Veränderung von Systemeigenschaften. Es können zusätzlich Argumente, die weitere Funktionen auslösen, mit angegeben werden. STAT «argument» STAT «cr» gibt den auf der Diskette freien Speicherplatz und den Schreibzustand an. (auch STAT «laufwerk»)

### **STAT USR:**

Die auf der Diskette angelegten Benutzerbereiche und der aktive Benutzerbereich werden in einer Übersicht ausgegeben.

### **STAT DEV:**

gibt eine Tabelle mit den Zuordnungen der Kanäle zu bestimmten Geräten aus.

### **STAT «Kanal»:=«Gerät»**

weist den möglichen Kanälen CON:, LST:, PUN: oder RDR: ein bestimmtes Gerät zu.

### **STAT DSK:**

ergibt eine Tabelle der Laufwerkseigenschaften der aktiven Laufwerke. (Speicherplatz, Sektoren etc.)

### **STAT «name.ext»**

liefert eine Übersicht über die Größe der angegebenen Dateien. Mehrfachnamen zulässig

### **STAT «name.ext» \$«merkmal»**

setzt Dateimerkmale.

### **\$R/W**

(read/write) die Datei kann sowohl gelesen als auch beschrieben werden.

### **\$R/O**

(read-only) Es kann nur aus der Datei gelesen werden. Schreiben führt zu

einer Fehlermeldung.

### \$DIR

Die Datei erscheint nach dem Befehl «DIR» mit im Inhaltsverzeichnis.

### \$SYS

Die Datei wird auf die Systemebene gesetzt und erscheint nicht mehr bei Ausgabe eines Inhaltverzeichnisses.

Alle Merkmale sind beliebig einsetzbar.

## Kanäle

CON: Konsolenkanal (Bildschirmgerät m. Tastatur)

LST: Druckerkanal (Drucker)

PUN: Lochstreifenstanzer (Fernschreiber)

RDR: Lochstreifenleser (Fernschreiber)

Prinzipiell sagt diese von Digital Research vorgesehene Kanal-Bezeichnung nichts über die tatsächliche Verwendung einzelner Kanäle aus. Sie können beliebig benutzt werden.

## Geräte

Den Kanälen können jeweils bestimmte Geräte zugeordnet werden.

z.B. LST:=TTY: LPT: UL1: UL2:

CON:=TTY: CRT: BAT: UC1:

## PIP

Mit diesem Programm besteht die Möglichkeit zum Informationsaustausch mit Peripheriegeräten (Diskettendateien und Kanäle (CON:, LST: etc.)). Die Hauptaufgabe dieses CP/M Befehls, der auch über eine Programmdatei abgewickelt wird «PIP.COM», ist das Kopieren von Dateien zwischen Disketten (Festplatte).

### Kopieren

Eine Datei wird durch die Eingabe: «Ziellaufwerk»:=«Herkunftslaufwerk» «name.ext» von der Diskette im Herkunftslaufwerk auf die Diskette im Ziellaufwerk kopiert. Es können Mehrfachnamen angegeben werden. B:=A\*.\* kopiert alle Dateien von A: nach B:.

### Drucken

Zum Ausdruck von ASCII-Dateien kann PIP Daten auf einen Kanal ausgeben. PIP LST:=«name.ext» schickt den Dateiinhalt auf den angeschlossenen Drucker.

### Wiederholungseingaben

Wird PIP ohne zusätzliche Angaben aufge rufen, so erscheint als Prompt ein «\*». Das ist die Bereitschaftsmeldung. Es können jetzt nach dem «\*» die Anweisungen eingegeben werden, ohne PIP jeweils von der Diskette zu laden.

### Optionen

PIP bietet die Möglichkeit, den Übertragungsprozeß zu verändern. Die Optionen werden in eckigen Klammern «[ ]» direkt nach der Befehlseingabe angegeben. PIP «Laufwerk»:=«name.ext» [«Optionen»]

|      |  |
|------|--|
| [V]  | Verify überprüft die Zielfile auf korrekte Übertragung. Nur bei Diskettendateien.                              |
| [E]  | Echo wiederholt alle gelesenen Zeichen auf dem Bildschirm. Nur bei ASCII-Dateien zu verwenden.                 |
| [Gn] | Get user liest die angegebene Datei aus dem in «n» bezeichneten User-Bereich.                                  |
| [B]  | Block mode transfer. Es wird das Software-Protokoll XON- XOFF (controll-S) gefahren.                           |
| [H]  | Hex data transfer. Es werden nur Daten im Intel-Hex- Format übertragen.  |
| [O]  | Object file transfer. Es wird über alle Control-Z (Dateiende) hinaus übertragen. Nur bei Dateizusammenfassung. |
| [N]  | Zeilenummerierung.   |
| [N2] | Zeilenummerierung mit führenden Nullen und TAB.  |
| [Pn] | Fügt alle «n» Zeilen einen Seitenvorschub ein.   |
| [W]  | Schreibt über schreibgeschützte Dateien (R/O) ohne Rückfrage.  |
| [R]  | Liest System-Dateien. Mit Stat veränderbar.  |

## SUBMIT

Mit dem Programm «SUBMIT.COM» wird unter CP/M die Möglichkeit geschaffen, gleiche, immer wieder kehrende Befehlsabläufe aus einer Textdatei (Stapeldatei) heraus auszuführen. Es ist eine Datei mit den einzelnen Befehlen, je ein Befehl (o. Programmaufruf) pro Zeile, zu erstellen, (keine Leerzeilen, «return» nach der letzten Zeile).

### Dateiname

Der Name der «Submit» Stapeldatei kann beliebig gewählt werden. Die Klassenbezeichnung muß «SUB» lauten.

### Aufruf

Die Ausführung einer Stapelverarbeitung erfolgt durch:

SUBMIT «name»

### Kommentare

In die Stapeldatei können neben den Befehlen auch Kommentare eingefügt werden, die auf dem Bildschirm während der Abarbeitung der Datei erscheinen. Kommentar- Zeilen müssen mit einem Zeichen («»,.: — ?\*[]) beginnen.

### Variable

werden in einer Stapeldatei durch das «\$»-Zeichen und einer fortlaufenden Nummer eingeführt.  
Variable = \$1..

Variablen-Eingaben erfolgen zusammen mit dem Aufruf der Submit- Datei.  
SUBMIT «name» «Eingabe für Variable 1»..

Zur Ergänzung der Submit-Funktion steht das Programm «XSUB.COM» zur Verfügung.

## XSUB

Mit XSUB wird die automatische Eingabe in laufende Programme, die mit der Submit-Funktion aufgerufen wurden, ermöglicht.

### Aufruf

XSUB muß zu Beginn in der Stapeldatei in einer eigenen Befehlszeile erscheinen.

### Programmeingaben über XSUB

Die von einem Programm erwarteten Eingaben sind in der entsprechenden Reihenfolge, je einer in einer Befehlszeile der Stapeldatei, anzugeben. Die Eingaben können auch wieder Variable sein, die beim Aufruf der gesamten Submit Funktion mit angegeben werden.

Beispiel: Erstellen einer Backup-Kopie Submit-Datei «BACKUP.SUB»

```
: **** Erstellen einer Backup-Kopie **** «return»
PIP B:=A:.* «return»
: **** Ende **** «return»
```

### Steuerzeichen

Bei Eingaben über die Tastatur sind folgende Steuerzeichen wirksam:

|           |  |
|-----------|--|
| Control X | Befehlszeile löschen                       |
| Control U | Eingabe rückgängig machen                  |
| Control E | Zeile beenden und nächste Zeile beginnen   |
| Control R | Befehlszeile abschließen                   |
| Control H | Cursor eine Position nach links            |
| Control C | Warmstart auslösen (nach Diskettenwechsel) |
| Control S | Pause                                      |
| Control Q | Start nach einer Pause                     |
| Control I | Tabulator                                  |

## DDT

(dynamic debugging tool) Dieses Programm ist ein dynamischer «Debugger», mit dem andere Programme schrittweise abgearbeitet werden können. Das ist oft notwendig, um die korrekte Funktion eines (selbst geschriebenen) Programmes zu untersuchen. Darüber hinaus können Programme disassembliert, einzelne Bytes (Speicher stellen) verändert oder Speicherbereiche im Hauptspeicher verschoben werden.

Aufruf von DDT: DDT «name.ext»

### I«name.ext»

Einen Dateinamen für eine Leseoperation vorgeben.

### R«versatz»

Die mit «l» angegebene Datei wird gelesen und an der Adresse (Originaladresse des

Programms + Versatz) abgelegt.

#### **G«start», «stopt», «stop2»..**

Das Programm wird ab «Start» ausgeführt mit Haltepunkten bei «Stop 1» etc.

#### **T«anzahl»**

Die angegebene «Anzahl» Programmschritte wird einzeln ausgeführt und der aktuelle Prozessorzustand angezeigt.

#### **U«anzahl»**

Wie T, nur der Prozessorzustand wird nicht angezeigt.

#### **X**

Der Prozessorzustand wird vollständig angezeigt.

#### **X«Register»**

Das angegebene Register wird angezeigt.

#### **D«anfang, ende»**

Der Speicherinhalt wird von «anfang» bis «ende» Hexadezimal (Sedezimal) und im ASCII-Format ausgegeben.

#### **L«anfang», «ende»**

Der Speicherinhalt wird in 8080-Assembler mnemonics ausgegeben (disassembliert).

#### **A«adresse»**

Es können mnemonicische Assembler-Befehle direkt eingegeben werden.

#### **S«adresse»**

Die angegebene Speicherstelle kann direkt (hexadezimal) verändert werden.

#### **F«anfang», «ende», «wert»**

Der Speicherbereich wird mit «wert» gefüllt.

#### **M«anfang», «ende», «ziel»**

Der Speicherbereich wird an die Adresse ab «Ziel» kopiert.

#### **H«wert 1», «wert 2»**

Die Summe und die Differenz zwischen Wert 1 und Wert 2 wird ermittelt und Hexadezimal ausgegeben.

## **ASM**

ASM ist der Standard-Assembler unter CP/M. Mit ihm können selbstgeschriebene Programme (in 8080-Assembler) in das Intel- Hex-Format übersetzt werden. Die Datei muß die Klassenbezeichnung (ext) «ASM» tragen.

#### **Aufruf**

ASM «name»

Erzeugte Dateien Die vom Assembler erzeugten Dateien tragen stets den Namen der «. ASM»-Datei, jedoch unterschiedliche Klassenbezeichnung.

PRN-Datei Hier wird das komplette Listing des Programmes abgelegt.

HEX-Datei In dieser Datei wird das Programm im Intel-Hex-Format abgelegt.

## LOAD

Mit dem Programm «Load. COM» wird das zuvor assemblierte Programm vom Intel-Hex-Format in Maschinensprache übersetzt und in einer «COM» Datei abgelegt.

Fehlermeldungen Der Bedienungsprozessor CCP (console command

Prozessor) meldet fehlerhafte Systemzustände oder Eingaben auf dem Bildschirm an.

«name»? Das gewählte Programm befindet sich nicht auf der

Diskette oder der Name enthält ungültige Zeichen.

NO FILE Die angegebene Datei befindet sich nicht auf der

Diskette.

FILE EXISTS Beim Umbenennen wurde ein neuer Name gewählt,

der bereits auf der Diskette vorliegt.

NO SPACE Auf der Diskette ist nicht mehr ausreichend Platz.

Bdos Err on «Laufwerk»: Select

1. keine Diskette im Laufwerk
2. LaufwerkTür nicht geschlossen
3. ungültiges Laufwerk gewählt

Bdos Err on «Laufwerk»: FILE R/O Die angewählte Datei ist schreib

geschützt. Mit STAT veränderbar.

Bdos Err on «Laufwerk»: BAD SECTOR Die Diskette ist beschädigt.

Bdos Err on «Laufwerk»: R/O Die Diskette ist schreibgeschützt.

1. Schreibschutzkerbe zugeklebt
2. kein Warmstart nach einem Diskettenwechsel

(Control-C) durchgeführt.

## Bestandteile von CP/M

Das Betriebssystem CP/M besteht aus dem eigentlichen System, das sich während der Ausführung im Hauptspeicher befindet, und einer Reihe von Hilfsprogrammen, die zusammen mit dem Betriebssystem auf der von Digital Research gelieferten Diskette untergebracht sind.

Das Betriebssystem besteht aus drei Teilen:

|  |
|--|
| CCP (console command Processor) - Standard         |
| BDOS (basic disk operating System) - Standard      |
| BIOS (basic input/output System) – Rechnerabhängig |

(Basic hat hier nichts mit der «Programmiersprache» zu tun, sondern ist im Sinne von «Basis» oder «Grundlegend» zu verstehen)

Residente Befehle sind die direkt im CCP enthaltenen DIR, ERA, REN, TYPE, SAVE und USER.

Externe Befehle werden aus einzelnen Programmen heraus ausgeführt. Dabei handelt es sich um:

|            |          |
|------------|----------|
| PIP.COM    | DDT.COM  |
| STAT.COM   | ASM.COM  |
| SUBMIT.COM | LOAD.COM |

zusätzliche Programme:

FORMAT.COM - zum Initialisieren (formatieren) einer neuen

Diskette

SYSGEN.COM - zum Kopieren von CP/M auf die Systemspuren

einer (neuen) Diskette

## ED

CP/M Befehl

Art: transienter Befehl

ED ist ein zeilenorientierter Texteditor. Das bedeutet, daß Sie jeden einzelnen Bearbeitungsschritt innerhalb von ED in Form von einzelnen Befehlen eingeben müssen (ähnlich wie bei EDLIN unter MS-DOS). Für jeden Bearbeitungsschritt müssen Sie ED also einen neuen Befehl geben. Eine Übersicht sämtlicher ED-Befehle siehe unten. Dieses Programm ist ein echter Dinosaurier und besitzt eine für heutige Verhältnisse sehr gewöhnungsbedürftige Bedienung. Wenn Sie größere Texte editieren wollen, nehmen Sie lieber das frei erhältliche Programm ZDE oder einen „anständigen“ Editor wie z.B.



WordStar

Die zu bearbeitende Textdatei muß in der Befehlszeile als Argument angegeben werden. Sie wird beim Starten von ED in einen Bearbeitungspuffer ins RAM geladen. Beim Beenden von ED wird der Text dann zurück in die Quelldatei geschrieben. Die ursprüngliche Datei wird hierbei jedoch nicht gelöscht, sondern bleibt als Sicherheitskopie unter dem alten Namen mit der Dateikennung .BAK bestehen. Falls Sie in der Befehlszeile jedoch eine Zielfile als Argument angegeben haben, wird stattdessen der Text dann in diese Zielfile geschrieben. Die Zielfile kann natürlich auch auf einem anderen Laufwerk liegen als die Quelldatei.

Beispiel:

ED MEINTEXT.TXT

Startet ED und lädt die Datei MEINTEXT.TXT in den Bearbeitungspuffer. Falls die Datei MEINTEXT.TXT noch nicht vorhanden ist, kann Sie neu erstellt werden. ED gibt in diesem Fall die Meldung „NEW FILE“ aus.

ED BEISPIEL.TXT B:NEU.TXT

Startet ED und lädt die Datei BEISPIEL.TXT in den Bearbeitungspuffer. Beim Beenden von ED wird die editierte Datei unter dem Namen NEU.TXT auf Laufwerk B gespeichert.

Übersicht der einzelnen ED-Befehle:

ED-Befehle

Beschreibung

n A

Lädt n Zeilen aus der Quelldatei in den Bearbeitungspuffer von ED

0 A

Lädt soviele Zeilen aus der Quelldatei, bis die Hälfte des Bearbeitungspuffers voll ist. (Das Zeichen vor dem „A“ ist eine Null, bitte nicht mit dem Buchstaben „O“ verwechseln)

# A

Lädt soviele Zeilen aus der Quelldatei, bis der Bearbeitungspuffer voll ist.

B -B

Positioniert den Textzeiger (character pointer) an den Anfang (B) oder an das Ende (-B) des Bearbeitungspuffers

n C -n C

Positioniert den Textzeiger (character pointer) um n Zeichen vorwärts (C) oder um n Zeichen rückwärts (-C).

n D -n D

Löscht n Zeichen vor (-D) oder hinter (D) der Cursorposition.

E

Beendet die Editierung. Die aktuelle Datei wird unter Ihrem angegebenen Namen auf der Diskette abgespeichert. Falls ein Zielname beim Aufruf von ED angegeben, wird dieser verwendet. Falls bereits eine alte Version der Datei mit gleichem Namen vorhanden ist, wird diese in eine Backupdatei umbenannt (Dateiendung .BAK). Nachdem dies geschehen ist, springt ED zurück zur CP/M-Eingabeaufforderung.

F s

Sucht die angegebene Zeichenkette s im gesamten Textpuffer.

H

Beendet die Editierung, speichert die neue Datei, ruft die neue Datei als neue Quelldatei auf und setzt mit dieser die Editierung fort.

I

Aufrufen des Einfügemodus für eine neue Textzeile hinter der aktuellen Cursorposition.

I s

Fügt die angegebene Zeichenkette s an der aktuellen Cursorposition ein und positioniert den Textzeiger anschließend hinter das letzte Zeichen der Zeichenkette s.

J s1 ^Z s2 ^Z s3

Nebeneinanderstellen von Zeichenketten, indem die erste Zeichenkette gesucht wird, die zweite an die erste angefügt wird und alle Zeichen bis zur dritten gelöscht werden. Das Zeichen ^Z steht für die Tastenkombination Control-Z.

n K -n K

Löscht ausgehend von der aktuellen Cursorposition die folgenden (K) oder vorstehenden (-K) n Zeilen.

n L -n L 0 L

Positioniert den Cursor um n Zeilen vorwärts (n L) oder rückwärts (-n L). Mit 0 L wird der Cursor an den Beginn der aktuellen Zeile gesetzt. (Das Zeichen vor dem „L“ ist eine Null, bitte nicht mit dem Buchstaben „O“ verwechseln)

n M command

Führt die Befehle „command“ n-mal aus.

n -n

Positioniert den Cursor um n Zeilen vorwärts (n) oder rückwärts (-n) und zeigt diese Zeile an.

n:

Positioniert den Cursor an den Beginn der Zeile n.

## : n command

Führt den Befehl „command“ bis zur Zeile n aus.

## N s

Sucht die angegebene Zeichenfolge s im Textpuffer.

## O

Rücksprung in die Originaldatei.

## n P -n P

Positioniert den Cursor jeweils 23 Zeilen weiter (= Bildschirmseitenlänge) und listet diese Zeilen auf dem Bildschirm auf.

## Q

Die Editierung der aktuellen Datei wird abgebrochen und ED wird verlassen, ohne die Datei zu speichern. Alle Änderungen am Text seit der letzten Speicherung gehen verloren.

## R

Einlesen der Datei X\$\$\$\$\$.LIB in den Textpuffer. (Weiß jemand was das zu bedeuten hat?)

## R Dateiname

Einlesen einer Datei in den Textpuffer. An der Stelle des Platzhalters „Dateiname“ geben Sie bitte den Namen der gewünschten Datei an..

## S s1 ^Z s2

Die Zeichenkette s1 wird durch die Zeichenkette s2 ersetzt. Das Zeichen ^Z steht für die Tastenkombination Control-Z

## n T -n T 0 T

Listet n Zeilen auf. Die Handhabung von n, -n und 0 erfolgt in Bezug auf die aktuelle Cursorposition so wie bei den anderen Befehlen. (Das Zeichen vor dem „T“ ist eine Null, bitte nicht mit dem Buchstaben „O“ verwechseln)

## U -U

Wandelt alle folgenden Kleinbuchstaben nach der Eingabe von U in Großbuchstaben um. Die Umwandlung wird mit -U beendet.

## V -V

Schaltet die Anzeige von Zeilenummern im Textpuffer mit der Eingabe von V ein. Diese Anzeige wird mit -V ausgeschaltet

## 0 V

Zeigt die verfügbare und die gesamte Größe des Textpuffers in Bytes dezimal an.

n W

Schreibt n Zeilen in die Übergangsdatei (Datei mit der Endung \$\$\$)

0 W

Schreibt soviele Zeilen in die Übergangsdatei, bis der Puffer halb voll ist. (Das Zeichen vor dem „W“ ist eine Null, bitte nicht mit dem Buchstaben „O“ verwechseln.)

n X

Kopiert die folgenden n Zeilen in die Datei X\$\$\$\$\$.LIB

n X Dateiname

Kopiert die folgenden n Zeilen in die Datei, welche anstelle von „Dateiname“ angegeben worden ist. Wurde der X-Befehl schon vorher verwendet, werden diese Zeilen daran angefügt.

0 X

Löscht die Datei X\$\$\$\$\$.LIB

0 X Dateiname

Löscht die Datei, welche anstelle von „Dateiname“ angegeben worden ist.

n Z

Unterbricht die laufende ED-Operation um n Sekunden.

Sie können innerhalb von ED natürlich auch mehrere Befehle gleichzeitig eingeben. Die einzelnen Befehle werden in der Eingabezeile mittels der Tastenkombination Control-Z voneinander getrennt.

From:  
<https://hc-ddr.hucki.net/wiki/> - **Homecomputer DDR**

Permanent link:  
<https://hc-ddr.hucki.net/wiki/doku.php/cpm/cpm22>



Last update: **2015/03/05 12:55**