Einige wichtige Funktionen des DOS-Interrupts 21hex

Wolfgang Kippels

19. September 2014

Inhaltsverzeichnis

1 Vorwort

Diese Datei gehört zu einem Assembler-Lehrgang, der hier zu finden ist:
http://dk4ek.de/lib/exe/fetch.php/as_lehrg.pdf

Unter DOS existieren viele verschiedene Interrupts, beispielsweise:

Interrupt 20hex: Mit diesem Interrupt kann ein Programm beendet werden. (Es sollte jedoch die Funktion **4Chex** des Interrupt **21hex** vorgezogen werden, da nur diese Funktion alle geöffneten Dateien automatisch schließt.)

Interrupt 22hex: Mit diesem Interrupt wird eine Adresse festgelegt, zu der nach Beendigung des Programmes gesprungen wird.

Interrupt 23hex: Dieser Interrupt wird aufgerufen, wenn die Tastenkombination **Strg-C** aufgerufen wird.

Interrupt 24hex: Dieser Interrupt wird automatisch beim Auftreten eines kritischen Fehlers aufgerufen.

Interrupt 25hex: Dieser Interrupt wird für absolutes Lesen verwendet.

Interrupt 26hex: Dieser Interrupt wird für absolutes Schreiben verwendet.

Interrupt 33hex: Dieser Interrupt ist für alle Funktionen mit der Maus zuständig.

Interrupt 67hex: Dieser Interrupt ist für EMS und XMS zuständig.

Der wichtigste Interrupt ist jedoch der Funktionen-Interrupt **21hex**. Er stellt fast alle Funktionen zur Verfügung, die man möglicherweise benötigt. Im folgenden werden nur die wichtigsten Funktionen dieses Interrupts dargestellt. Die Funktionsnummer muss beim Aufruf des Interrupts **21hex** immer im Register **AH** stehen. Je nach Funktion muss dann noch eine Unterfunktionsnummer in **AL** stehen. Alle anderen relevanten Register und ihre Bedeutung sind bei den jeweiligen Funktionen beschrieben. Dabei bedeutet:

Eingabe: Die angegebenen Register müssen **vor** dem Funktionsaufruf auf die angegebenen Werte eingestellt werden.

Ausgabe: Diese Werte liefert die Funktion **nach** ihrem Aufruf an den aufrufenden Programmteil zurück. Bei einigen Funktionen kann ein Fehler auftreten. Das wird ggf. durch ein gesetztes Carry-Flag (**CF**) angezeigt. Dann steht im Register **AX** eine Fehlernummer. Ist das Carry-Flag gelöscht, lief die Funktion ordnungsgemäß ab.

2 Funktion 01h: Zeicheneingabe mit Echo

Eingabe: **AH**=01h **Ausgabe:** **AL**=Zeichencode

- Durch die Möglichkeit der Umleitung der Ein- und/oder Ausgabe können die Geräte zur Ein- bzw. Ausgabe von Funktionsaufruf zu Funktionsaufruf unterschiedlich sein.
- Das Zeichen wird eingelesen und gleichzeitig auf der Standard-Ausgabe ausgegeben.
- Beim Drücken von **Strg-C** wird der Interrupt **23h** ausgelöst.
- Es wird so lange gewartet, bis ein Zeichen verfügbar ist.
- Auch Sondertasten, die keinen ASCII-Code erzeugen, können mit dieser Funktion gelesen werden. Für einen solchen „erweiterten Code“ liefert die Funktion zunächst den Wert **00h** (oder auch **E0h**) zurück. Erst beim erneuten Aufruf der Funktion liefert die Funktion den eigentlichen Funktionscode zurück. Daher sollte man der Funktion **10h** des BIOS-Interruptes **16h** vor dieser Funktion für Tastaturabfragen den Vorzug geben.

3 Funktion 02h: Zeichen ausgeben

Eingabe: **AH**=02h
DL=Zeichencode
Ausgabe: keine

- Durch die Möglichkeit der Umleitung der Ein- und/oder Ausgabe können die Geräte zur Ein- bzw. Ausgabe von Funktionsaufruf zu Funktionsaufruf unterschiedlich sein.
- Beim Drücken von **Strg-C** wird der Interrupt **23h** ausgelöst.
- Ist das Ausgabegerät der Bildschirm oder der Drucker, so werden Bildschirmsteuerzeichen (**CR**, **LF** usw.) ausgeführt, bei Ausgabe auf ein anderes Gerät jedoch nicht.
- Die Funktion **0Eh** des BIOS-Interrupt **10h**, die ebenfalls eine Zeichenausgabe bewirkt, arbeitet erheblich schneller als diese DOS-Funktion. Wenn nicht die Möglichkeit der Umleitung benötigt wird, ist die Funktion des BIOS-Interrupt also zu bevorzugen.

4 Funktion 06h: Zeichenein- und -ausgabe direkt

Eingabe:	**AH**=06h **DL**=Zeichencode oder Leseflag	**Ausgabe:**	nur bei Zeicheneingabe: Zero-Flag zeigt Eingabestatus an **AL** enthält Zeichencode

- Ein Wert von **FFh** im **DL**-Register beim Aufruf der Funktion zeigt an, dass ein Zeichen eingelesen werden soll. Jeder andere Wert wird als Zeichencode interpretiert, der ausgegeben werden soll.
- Ein gesetztes Zero-Flag nach der Zeichen**eingabe** (also Aufruf der Funktion mit **DL=FFh**) zeigt an, dass kein Zeichen bereitsteht. Ein gelöschtes Zero-Flag zeigt an, dass der Zeichencode in **AL** gültig ist.
- Es wird **keine** Überprüfung auf **Strg-C** vorgenommen.

5 Funktion 07h: Zeicheneingabe direkt

Eingabe:	**AH**=07h **AL**=Zeichencode	**Ausgabe:**	keine

- Ein Zeichen wird eingelesen, aber nicht auf der Standard-Ausgabe ausgegeben.
- Es wird so lange gewartet, bis ein Zeichen verfügbar ist.
- Es wird **keine** Überprüfung auf **Strg-C** vorgenommen.
- Auch Sondertasten, die keinen ASCII-Code erzeugen, können mit dieser Funktion gelesen werden. Für einen solchen „erweiterten Code“ liefert die Funktion zunächst den Wert **00h** (oder auch **E0h**) zurück. Erst beim erneuten Aufruf der Funktion liefert die Funktion den eigentlichen Funktionscode zurück. Daher sollte man der Funktion **10h** des BIOS-Interruptes **16h** vor dieser Funktion für Tastaturabfragen den Vorzug geben.

6 Funktion 08h: Zeicheneingabe

Eingabe: **AH**=08h **Ausgabe:** **AL**=Zeichencode

- Durch die Möglichkeit der Umleitung der Ein- und/oder Ausgabe können die Geräte zur Ein- bzw. Ausgabe von Funktionsaufruf zu Funktionsaufruf unterschiedlich sein.
- Das Zeichen wird eingelesen, aber **nicht** auf der Standard-Ausgabe ausgegeben.
- Beim Drücken von **Strg-C** wird der Interrupt **23h** ausgelöst.
- Es wird so lange gewartet, bis ein Zeichen verfügbar ist.
- Auch Sondertasten, die keinen ASCII-Code erzeugen, können mit dieser Funktion gelesen werden. Für einen solchen „erweiterten Code“ liefert die Funktion zunächst den Wert **00h** (oder auch **E0h**) zurück. Erst beim erneuten Aufruf der Funktion liefert die Funktion den eigentlichen Funktionscode zurück. Wenn man die Abbruchmöglichkeit des Programms mit **Strg-C** nicht benötigt, sollte man der Funktion **10h** des BIOS-Interruptes **16h** vor dieser Funktion für Tastaturabfragen den Vorzug geben.

7 Funktion 09h: Ausgabe einer Zeichenkette

Eingabe: **AH**=09h **Ausgabe:** keine
DS:DX=Adresse der Zeichenkette

- Die auszugebende Zeichenkette muss mit einem Dollar-Zeichen $ als Stringendemarke abgeschlossenen sein. Das Zeichen $ wird dabei **nicht** mit ausgegeben.
- Steuerzeichen (z.B. Zeilenvorschub oder Tabulator) werden als solche ausgeführt und **nicht** in ihrer symbolischen Schreibweise dargestellt.

8 Funktion 0Ah: Eingabe einer Zeichenkette

Eingabe: **AH**=0Ah
DS:DX=Adresse des Textpuffers

Ausgabe: keine

- Es wird eine Texteingabe erwartet, die mit <Enter> abgeschlossen wird.
- Im 1. Byte des Textpuffers muß die maximale Länge des einzugebenden Strings (ohne das abschließenden <Enter>) übergeben werden. Im String muss aber auch für dieses Enterzeichen noch Platz sein.
- Es werden so lange Zeichen eingelesen, bis das Zeichen **0Dh** (Entertaste) gelesen wird.
- Wird bei der Eingabe die maximale Zeichenzahl überschritten, so gibt DOS einen Piepston aus und nimmt nur noch das Zeichen **0Dh** (Enterzeichen) an.
- Bei Eingabe von der Tastatur kann die Eingabe mit den Cursortasten und/oder der **Entf**-Taste editiert werden.
- Falls erweiterte Zeichen-codes (Funktionstasten) eingegeben werden, belegen diese im Eingabepuffer zwei Bytes: eine Null gefolgt von dem eigentlichen Code.
- Wird **Strg-C** eingegeben, so wird ein Interrupt **23h** ausgelöst.
- Der Text steht anschließend ab dem 3. Byte im Textpuffer, im 2. Byte die tatsächliche Länge des String (ohne abschließendes <Enter>).

9 Funktion 0Ch: Eingabepuffer löschen und Funktion aufrufen

Eingabe: **AH**=0Ch
AL=Funktionsnummer

Ausgabe: **AL**=Löschflag

- Die Funktion löscht zuerst den Eingabepuffer und ruft anschließend eine neue DOS-Funktion auf. Erlaubt sind die Funktionen **01h**, **06h**, **07h**, **08h** oder **0Ah**.
- Wird eine andere als diese aufgeführten Nummern in **AL** eingetragen, dann wird nur der Puffer gelöscht. Eine weitere Funktion wird nicht aufgerufen.
- Möglicherweise müssen – je nach Art der Funktion – weitere Parameter übergeben werden. Das kann bei der jeweiligen Funktion nachgesehen werden.
- Das Löschflag gibt an, ob die Löschfunktion ordnungsgemäß ausgeführt wurde. Es ist nur dann relevant, wenn keine der Eingabefunktionen als Anschlussfunktion aufgerufen wurde. Mit **AL=00h** zeigt es ein erfolgreiches Löschen an.

10 Funktion 0Eh: Laufwerk auswählen

Eingabe: **AH**=0Eh
DL=Laufwerkcode

Ausgabe: Anzahl der installierten Blocktreiber (Laufwerke)

- Laufwerkcodes:

  00h: Laufwerk **A**

  01h: Laufwerk **B**

  02h: Laufwerk **C** usw.

- Der zurückgelieferte Wert für die Anzahl der Laufwerke ist nicht verlässlich. Es kann eine größere Zahl zurückgeliefert werden.

11 Funktion 19h: Aktuelles Laufwerk bestimmen

Eingabe: **AH**=19h

Ausgabe: **DL**=Laufwerkcode

- Laufwerkcodes:

  00h: Laufwerk **A**

  01h: Laufwerk **B**

  02h: Laufwerk **C** usw.

12 Funktion 25h: Interrupt-Vektor setzen

Eingabe: **AH**=25h
AL=Interrupt-Nummer
DS:DX=Neue Interrupt-Adresse

Ausgabe: keine

13 Funktion 2Ah: Datum ermitteln

Eingabe: **AH**=2Ah

Ausgabe: **AL**=Wochentag
CX=Jahr
DH=Monat
DL=Tag des Monats

- Für den Wochentag gilt folgende Codierung:

  00h: Sonntag

  02h: Montag usw.

- Das Jahr ist als Offset zu 1980 abgelegt. Der Wert **00h** bedeutet demnach 1980, der Wert **01h** 1981, usw.

14 Funktion 2Ch: Zeit ermitteln

Eingabe: **AH**=2Ch

Ausgabe: **CH**=Stunde
CL=Minute
DH=Sekunde
DL=Hundertstelsekunde

- Die Zeit wird in 24-Stunden-Format ausgegeben.
- Je nach Hardware der Rechner können keine Hundertstelsekunden bestimmt werden. Der Rückgabewert in **DL** ist dann ein zufälliger Wert.

15 Funktion 30h: DOS-Versionsnummer ermitteln

Eingabe: **AH**=30h
AL=Leseflag

Ausgabe: **AH**=Nebenversionsnummer
BH=OEM-Seriennummer
AL=Hauptversionsnummer
BL:CX=24-Bit-Benutzernummer

- Das Leseflag muss den Wert 00h beinhalten.
- Ist das Carry-Flag gesetzt, so liegt ein Fehler vor. Der einzig mögliche Fehler-Code ist 02h: ein ungültiger Landes-Code.

16 Funktion 31h: Programm im Speicher behalten

Eingabe: **AH**=31h
AL=Return-Code
DX=Anzahl der belegten Paragraphen

Ausgabe: keine

- Der Return-Code ist der Code, den das Programm an das aufrufende Programm zurückgibt (Errorlevel).
- Ein Paragraph ist ein Speicherblock von 16 Bytes. Der Speicherplatz in Bytes, der reserviert werden soll, muss durch 16 geteilt werden, bei einem Rest noch 1 addiert werden und in **DX** eingetragen werden, bevor die Funktion aufgerufen wird.

17 Funktion 34h: DOS-Aktiv-Byte bestimmen

Eingabe: **AH**=34h

Ausgabe: **ES:BX**=Adresse des Aktiv-Bytes

- Besitzt das Aktiv-Byte den Wert **00h**, so ist **keine** DOS-Funktion aktiv, ansonsten enthält das Byte die Anzahl der aktiven Funktionen.

18 Funktion 35h: Interrupt-Vektor ermitteln

Eingabe: **AH**=35h
AL=Interrupt-Nummer

Ausgabe: **ES:BX**=Adresse des Interrupt-Vektors

19 Funktion 39h: Unterverzeichnis erstellen

Eingabe: **AH**=39h
DS:DX=Adresse des Pfadnamens

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**

- Der String des Pfadnamens muss mit dem Byte 00h abgeschlossen sein.
- Mögliche Fehlercodes:

  03h: Pfad oder Teil des Pfades nicht gefunden

  05h: Zugriff verweigert

20 Funktion 3Ah: Unterverzeichnis löschen

Eingabe: **AH**=3Ah
DS:DX=Adresse des Pfadnamens

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**

- Der String des Pfadnamens muss mit dem Byte 00h abgeschlossen sein.
- Mögliche Fehlercodes:

  03h: Pfad oder Teil des Pfades nicht gefunden

  05h: Zugriff verweigert

  10h: Das Verzeichnis ist das **aktuelle** Verzeichnis

21 Funktion 3Bh: Aktuelles Verzeichnis wechseln

Eingabe: **AH**=3Bh
DS:DX=Adresse des Pfadnamens

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**

- Der String des Pfadnamens muss mit dem Byte 00h abgeschlossen sein.
- Mögliche Fehlercodes:

  03h: Pfad oder Teil des Pfades nicht gefunden

22 Funktion 3Ch: Neue Datei anlegen

Eingabe:	**AH**=3Ch **DS:DX**=Adresse des Dateinamens **CX**=Attribut-Code	**Ausgabe:**	Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**. Wenn kein Fehler: **AX**=Handle-Nummer

- Der String des Pfadnamens muss mit dem Byte 00h abgeschlossen sein.
- Mögliche Fehlercodes:

  03h: Pfad oder Teil des Pfades nicht gefunden

  04h: zu viele geöffnete Dateien

  05h: Zugriff verweigert

- Wenn kein Fehler vorliegt, so wird in **AX** eine gültige Handle-Nummer (auch *File-Handle* genannt) übergeben. Bei allen weiteren Zugriffen auf die Datei muss diese Nummer verendet werden.
- Wenn eine Datei mit dem angegebenen Namen bereits existiert, erfolgt **keine Fehlermeldung**! Diese Datei wird überschrieben. Ist das **nicht** erwünscht, sollte man die Funktion **5Bh** verwenden.
- Für Attributcode in **CX** siehe nachfolgende Tabelle:

Bit-Nr.	Attribut
0	Die Datei ist schreibgeschützt.
1	Die Datei soll versteckt werden.
2	Die Datei ist eine System-Datei.

23 Funktion 3Dh: Datei öffnen

Eingabe: **AH**=3Dh
DS:DX=Adresse des Dateinamens
AL=Zugriffs-Code

Ausgabe: Carry-Flag zeigt Fehler an,
dann: Fehlercode in **AX**
wenn kein Fehler:
AX=Handle-Nummer

- Der String des Pfadnamens muss mit dem Byte 00h abgeschlossen sein.
- Mögliche Fehlercodes:

  02h: Datei nicht gefunden

  03h: Pfad oder Teil des Pfades nicht gefunden

  04h: Zu viele geöffnete Dateien

  05h: Zugriff verweigert, z.B. „Diskette schreibgeschützt“

  0Ch: Ungültiger Zugriff

- Zugriff-Codes in **AL** (siehe nachfolgende Tabelle)

Tabelle der Zugriffscodes:

Bit-Nr.	Bedeutung
0-1	Dateiattribut: **00b:** Datei darf **nur gelesen** werden. **01b:** Datei darf **nur geschrieben** werden. **10b:** Datei darf **gelesen und geschrieben** werden.
2-3	Muss immer Null sein.
4-6	Nur bei File-Sharing im Netzwerk (sonst Null): **000b:** Jeder Prozess darf auf die Datei zugreifen. **001b:** Nur der aktuelle Prozess darf auf die Datei zugreifen. **010b:** Ein anderer Prozess darf nur lesen. **011b:** Ein anderer Prozess darf nur schreiben. **100b:** Ein anderer Prozess darf schreiben und lesen.
7	**0b:** Auch Unterprogramme haben Zugriff. **1b:** Nur aktueller Prozess hat Zugriff.

24 Funktion 3Eh: Datei schließen

Eingabe: **AH**=3Eh
BX=Handle-Nummer

Ausgabe: Carry-Flag zeigt Fehler an,
dann: Fehlercode in **AX**

- Möglicher Fehlercode:

  06h: Handle ungültig

25 Funktion 3Fh: Lesen aus Datei

Eingabe: **AH**=3Fh
BX=Handle-Nummer
CX=Anzahl der zu lesenden Bytes
DS:DX=Adresse des Lese-Puffers

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**
Wenn kein Fehler:
AX=Anzahl der gelesenen Bytes

- Es kann auch aus einem der Standard-Handles gelesen werden, z.B. **00h**=Standard-Ein-/Ausgabe. Standard-Handles sind **immer geöffnet**.
- Mögliche Fehlercodes:

  05h: Zugriff verweigert

  06h: Handle ungültig

26 Funktion 40h: Schreiben in Datei

Eingabe: **AH**=40h
BX=Handle-Nummer
CX=Anzahl der zu schreibenden Bytes
DS:DX=Adresse des Schreib-Puffers

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**
Wenn kein Fehler:
AX=Anzahl der geschriebenen Bytes

- Es kann auch in eins der Standard-Handles geschrieben werden, z.B. **00h**=Standard-Ein-/Ausgabe. Standard-Handles sind **immer geöffnet**.
- Mögliche Fehlercodes:

  05h: Zugriff verweigert

  06h: Handle ungültig

27 Funktion 41h: Datei löschen

Eingabe: **AH**=41h
DS:DX=Adresse des Dateinamens

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**

- Mögliche Fehlercodes:

  02h: Datei nicht gefunden

  03h: Pfad oder Teil des Pfades nicht gefunden

  05h: Zugriff verweigert

28 Funktion 42h: Datei-Zeiger bewegen

Eingabe: **AH**=42h
BX=Handle-Nummer
AL=Bewegungsmodus
CX=Höherwertiges Wort der 32-Bit-Differenz
DX=Niederwertiges Wort der 32-Bit-Differenz

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**, sonst:
AX=niederwertiges Wort der neuen 32-Bit-Zeigerposition
DX=höherwertiges Wort der neuen 32-Bit-Zeigerposition

Bewegungsmodus:

Code:	Bedeutung:
00h	Bewegung des Zeigers relativ zum Dateianfang
01h	Bewegung des Zeigers relativ zur aktuellen Position
02h	Bewegung des Zeigers relativ zum Dateiende

- Mögliche Fehlercodes:

  01h: Bewegungsmodus ungültig

  06h: Handle ungültig

29 Funktion 43h: Datei-Attribut lesen oder ändern

Eingabe: **AH**=43h
AL=Modus
CX=Dateiattribut
DS:DX=Adresse des Dateinamens

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**, sonst:
CX=Dateiattribut

Dateiattribut:

Bit-Nr.:	Bedeutung bei gesetztem Bit:
0	Schreibgeschützte Datei
1	Versteckte Datei
2	System-Datei
3	Volume-Name (Datenträgername)
4	Unterverzeichnis
5	Archiv (Änderung nach letztem Backup)

Modus (in AL):

Nummer:	Aktion:
00h:	„Attribut lesen“
01h:	„Attribut setzen“

- Mögliche Fehlercodes:

  01h: Falscher Modus

  02h: Datei existiert nicht

  03h: Pfad nicht gefunden

  05h: Zugriff verweigert

30 Funktion 47h: Absolutes (aktuelles) Verzeichnis ermitteln

Eingabe: **AH**=47h
DL=Laufwerknummer
DS:SI=Adresse des Pfad-Puffers

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**

- Laufwerknummer:

  00h: aktuelles Laufwerk

  01h: Laufwerk A

  02h: Laufwerk B

  03h: Laufwerk C usw.

- Der Pfadname wird als ASCIIZ-String[1] abgelegt.

- Er enthält weder eine vorangehende Laufwerkbezeichnung noch einen vorangehenden Rückstrich \. Ist das Stammverzeichnis das aktuelle Verzeichnis, wird ein Leerstring zurückgegeben.
  Beispiel:

  Verzeichnis: `C:\TEXTE\BRIEFE`

  Ausgabe: `TEXTE\BRIEFE`

- Einziger möglicher Fehlercode:

  0Fh: Unbekanntes Laufwerk

[1] Ein ASCIIZ-String ist ein String, der mit einem Byte mit der Codierung 00h als „Stringendemarke“ abgeschlossen ist.

31 Funktion 48h: Speicherbereich zuweisen (anfordern)

Eingabe:	**AH**=48h **BX**=Paragraphenzahl	**Ausgabe:**	Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX** und **BX**=Zahl der verfügbaren Paragraphen sonst: **AX**=Speicher-Segment-Adresse

- Wenn kein Fehler auftrat (Carry-Flag gelöscht), steht in **AX** die Segmentadresse des zugewiesenen Speicherblockes.
- Mögliche Fehler:

  07h: MCBs sind zerstört.

  08h: Zu wenig Speicherplatz verfügbar.
- Falls kein ausreichender Speicherplatz vorhanden ist (Fehlercode 08h), liefert die Funktion in **BX** die Gesamtanzahl der verfügbaren Paragraphen[2].

32 Funktion 49h: Speicherbereich freigeben

Eingabe:	**AH**=49h **ES**=Segment-Adrese des Speicherbereiches	**Ausgabe:**	Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**

- Nur über Funktion 48h zugewiesener Speicherbereich kann auch wieder freigegeben werden.
- Mögliche Fehler:

  07h: MCBs sind zerstört

  09h: Falsche Segmentadresse

[2]Die „Paragraphenzahl“ ist die Anzahl der Bytes dividiert durch 16.

33 Funktion 4Ah: Speicherblockgröße verändern

Eingabe: **AH**=4Ah
BX=Paragraphenzahl
ES=Segment-Adrese des Speicherbereiches

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX** und **BX**=Zahl der verfügbaren Paragraphen

- Falls kein ausreichender Speicherplatz verfügbar ist (Fehlercode 08h), liefert die Funktion in **BX** die Gesamtanzahl der verfügbaren Paragraphen[3].
- Mögliche Fehler:

  07h: MCBs sind zerstört

  08h: Zu wenig Speicherplatz

  09h: Falsche Segmentadresse

34 Funktion 4Bh, Unterfunktion 00h: Unterprogramm laden und starten

Eingabe: **AH**=4Bh
AL=00h
DS:DX=Adresse des Dateinamens
ES:BX=Adresse des Parameterblockes

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**

[3]Die „Paragraphenzahl“ ist die Anzahl der Bytes dividiert durch 16.

Aufbau des Parameterblockes:

Byte-Nr.:	Bedeutung:
00h-01h	Segment-Adresse des Environments
02h-03h	Offset-Adresse des Kommando-Strings
04h-05h	Segment-Adresse des Kommando-Strings
06h-07h	Offset-Adresse des ersten FCBs
08h-09h	Segment-Adresse des ersten FCBs
0Ah-0Bh	Offset-Adresse des zweiten FCBs
0Ch-0Dh	Segment-Adresse des zweiten FCBs

- Das Environment ist der Parameterblock, in dem die aktuell mit dem SET-Befehl gesetzten Systemparameter zu finden sind.
- Der Kommando-String ist die Zeichenkette, die einem Programm mit dem Aufruf zusammen übergeben werden soll. Das erste Byte des Kommando-Strings gibt die Länge des String (einschließlich des abschließenden **0Dh**=<Enter>, aber ohne das Längenbyte selbst) an.
- Wird als Environment-Adresse der Wert 0 übergeben, so soll das aufzurufene Programm das gleiche Environment verwenden, wie das aufrufende Programm.
- Die beiden FCBs[4], deren Adressen in der Parametertabelle übergeben werden, werden in die entsprechenden Felder des PSP des aufgerufenen Programms kopiert.
- Man kann die Funktion nur nutzen, wenn genügend Speicherplatz zur Verfügung steht. Dieser muß vor der Benutzung ggf. erst freigemacht werden (siehe Funktion 49h).
- DOS generiert für das aufgerufene Programm automatisch einen PSP.
- mögliche Fehler:

  01h: Falsche Funktionsnummer

  02h: Datei nicht gefunden

  03h: Pfad nicht gefunden

  04h: zu viele offene Dateien

  05h: Zugrif verweigert

  08h: Zu wenig Speicherplatz

  0Ah: Falsches Environment

  0Bh: Falsches Format

[4]FCB steht für *File-Control-Block*. FCBs wurden eigentlich nur in ganz frühen DOS-Versionen zur Verwaltung von Dateien verwendet. Nur aus Kompatibilitätsgründen werden sie noch unterstützt.

35 Funktion 4Bh, Unterfunktion 01h: Programm laden (nicht starten)

Eingabe: **AH**=4Bh
AL=01h
DS:DX=Adresse des Dateinamens
ES:BX=Adresse des Parameterblockes

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**

Aufbau des Parameterblockes:

Byte-Nr.:	Bedeutung:
00h-01h	Segment-Adresse des Environments
02h-03h	Offset-Adresse des Kommando-Strings
04h-05h	Segment-Adresse des Kommando-Strings
06h-07h	Offset-Adresse des ersten FCBs
08h-09h	Segment-Adresse des ersten FCBs
0Ah-0Bh	Offset-Adresse des zweiten FCBs
0Ch-0Dh	Segment-Adresse des zweiten FCBs
0Eh-11h	Start-Adresse des Programms (*)
12h-15h	Stack-Adresse des Programms (*)

- DOS lädt das gewünschte Programm und erzeugt einen passenden PSP. Das Programm wird aber nicht gestartet.
- (*) In den Feldern 0Eh bis 15h des Parameterblockes stehen nach dem Aufruf der Funktion die Adressen, die DOS beim Laden des Programms verwendet.
- mögliche Fehler:

  01h: Falsche Funktionsnummer

  02h: Datei nicht gefunden

  03h: Pfad nicht gefunden

  04h: zu viele offene Dateien

  05h: Zugrif verweigert

  08h: Zu wenig Speicherplatz

  0Ah: Falsches Environment

  0Bh: Falsches Format

36 Funktion 4Bh, Unterfunktion 03h: Unterprogramm als Overlay laden

Eingabe: **AH**=4Bh
AL=03h
DS:DX=Adresse des Dateinamens
ES:BX=Adresse des Parameterblockes

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**

Aufbau des Parameterblockes:

Byte:	Bedeutung:
00h-01h	Segment-Adresse des Overlays
02h-03h	Relozierungswert

- Die Segment-Adresse des Overlays ist die Adresse, ab der das Programm geladen werden soll.
- Der Relozierungswert ist der Wert, der bei EXE-Dateien dazu verwendet wird, die absolute Adressangaben innerhalb des eigenen Programms auf die veränderte Segmentadresse umzurechnen. Bei COM-Dateien muß der Wert 00h eingegeben werden.
- Das Programm wird als Overlay geladen. Das heißt, der Speicherplatz muß nicht zuvor reserviert werden, sondern muß innerhalb des Adressbereiches des aufrufenden Programmes liegen.
- Die Funktion wird meist dazu verwendet um innerhalb eines Programmes kleinere Unterprogramme nachzuladen.
- mögliche Fehler:

  01h: Falsche Funktionsnummer

  02h: Datei nicht gefunden

  03h: Pfad nicht gefunden

  04h: zu viele offene Dateien

  05h: Zugrif verweigert

  08h: Zu wenig Speicherplatz

37 Funktion 4Bh, Unterfunktion 05h: Programmausführung setzen

Eingabe: **AH**=4Bh
AL=05h
DS:DX=Adresse des Parameterblockes

Ausgabe: keine

Aufbau des Parameterblockes:

Offset:	Datentyp:	Bedeutung:
00h	Wort	Reserviert, muss 00h sein
02h	Wort	Dateityp: 0000h=COM-Datei 0001h=EXE-Datei 0002h=Overlay-Datei
04h	Doppelwort	Adresse des Programmnamens
08h	Wort	Segment-Adresse des PSP
0Ah	Doppelwort	Startadresse
0Eh	Doppelwort	Programmgröße (in Byte)

- Die Funktion dient dazu, dem Betriebssystem die Namen von Programmen mitzuteilen, die über programmeigene Laderoutinen geladen werden sollen.
- DOS sperrt die Adressleitung 20 bei diesem Funktionsaufruf. Programme, die auf Speicher über 1MByte zugreifen möchten, müssen diese Leitung wieder freigeben.

38 Funktion 4Ch: Programm beenden

Eingabe: **AH**=4Ch
AL=Returncode

Ausgabe: keine

- Der Returncode kann vom Mutterprogramm mit der Funktion **4Dh** oder in einer Batch-Datei mit „Errorlevel“ abgefragt werden.
- Üblicherweise wird **AL**=00h gesetzt, wenn ein fehlerfreies Beenden des Programms angezeigt werden soll. Die Werte können aber im Prinzip nach Belieben verwendet werden.
- Alle offenen Dateien werden geschlossen, die drei Interruptvektoren im PSP (Interrupts 22h, 23h und 24h) werden nach dem Funktionsaufruf in die entsprechenden Systemspeicherstellen übertragen.

39 Funktion 4Dh: Beendigungscode ermitteln

Eingabe: **AH**=4Dh **Ausgabe:** **AX** =Umkehr-Ende-Code

- **AH** enthält die Art der Programmbeendigung eines „Tochterprogramms“ (siehe untenstehende Tabelle), **AL** enthält den vom Tochterprogramm übergebenen Return-Code.

Bedeutung des Wertes in AH:

Wert:	Bedeutung:
00h	Normales Ende durch Programm selbst
01h	Programm wurde durch Strg-Pause oder Strg-C abgebrochen
02h	Programm wurde durch Fehler beim Gerätezugriff abgebrochen.
03h	Programm wurde durch Funktion 31h (bleibe resident) beendet.

40 Funktion 4Eh: Ersten Eintrag finden

Eingabe: **AH**=4Eh
CX=Datei-Attribut
DS:DX=Adresse des Dateinamens

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**

- Das Datei-Attribut kann dazu verwendet werden, auf bestimmte Besonderheiten im Dateinamen hinzuweisen. Mit dem Code für das Datei-Attribut kann man auch nach Unterverzeichnissen, Datenträgernamen oder besonderen Dateien suchen. Mögliche Attribute siehe Funktion **43h**.

- Der Dateiname muß als ASCIIZ-String[5] vorliegen. Der Dateiname kann auch ein kompletter Suchpfad sein, er darf also eine Gerätebezeichnung, einen Pfadnamen und einen Dateinamen inclusive Wildcards (? und *) enthalten. Ist das Carry-Flag gelöscht, dann werden in den ersten 44 Bytes der DTA Informationen über die Datei abgelegt. Die nachfolgende Tabelle stellt die Struktur der Parametertabelle in der DTA dar, darunter die Tabellen für das Format von Datum und Zeit.

Struktur der Parametertabelle:

Byte:	Bedeutung:
00h-14h	Reserviert für DOS
15h	Attribut des Eintrags
16h-17h	Erstellungszeit der Datei
18h-19h	Erstellungdatum der Datei
1Ah-1Dh	Größe der Datei (low/high)
1Eh-2Bh	Dateiname, mit 00h abgeschlossen

[5]Ein ASCIIZ-String ist ein String, der mit einem Byte mit der Codierung 00h als „Stringendemarke“ abgeschlossen ist.

Format des Datums:

Bits:	Bedeutung:
0-4	Tag des Monats (1 - 31)
5-8	Monat (1 - 12)
0-15	Jahr (als Offset zu 1980)

Format der Zeit:

Bits:	Bedeutung:
0-4	Sekunden /2 (3 entspricht 6 Sekunden)
5-10	Minuten
11-15	Stunden

- Die Funktion kann nur verwendet werden, um eine einzelne Datei (ohne Wildcards) oder die erste Datei aus einer Gruppe von Dateien (also Datei mit Wildcards) zu suchen.
- Für weiteres Suchen (Dateiname mit Wildcards) muss die Funktion **4Fh** verwendet werden.
- Mögliche Fehler:

  03h: Pfad nicht gefunden

  12h: Keine (weiteren) Dateien im Verzeichnis, auf die die Suchbedingung zutrifft.

41 Funktion 4Fh: Weiteren Eintrag finden

Eingabe: **AH**=4Fh

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**

- Die Funktion kann nur aufgerufen werden, nachdem bereits mit der Funktion **4Eh** gesucht worden ist. Sie kann jedoch mehrfach nacheinander aufgerufen werden, um noch weitere passende Dateien zu suchen.
- Mögliche Fehler siehe bei Funktion **4Eh**, ebenso die Ergebnisse in der DTA.

42 Funktion 50h: PSP-Adresse setzen

Eingabe: **AH**=50h
BX=Neue Segment-Adresse des PSP

Ausgabe: keine

- Die Funktion legt einen neuen PSP für ein laufendes Programm fest.

43 Funktion 51h: PSP-Adresse ermitteln

Eingabe: **AH**=51h **Ausgabe:** **BX**=Segment-Adresse des PSP

- Die Funktion ermittelt die Adresse des PSP für das aktuell laufende Programm, siehe auch Funktion **62h**.

44 Funktion 52h: DIB-Adresse ermitteln

Eingabe: **AH**=52h **Ausgabe:** **ES:BX**=Adresse des DIB

- Mit dieser Funktion wird die Adresse des **DIB**(=DOS-Informations-Block) festgestellt.
- **Achtung:** Der erste Eintrag im DIB steht schon 4 Bit **vor** der zurückgegebenen Adresse!
- Nachfolgend ist der Aufbau des DOS-Informations-Blockes dargestellt.

Aufbau des DOS-Informations-Blockes:

Offset:	Datentyp:	Inhalt:
-4h	Doppelwort	Zeiger auf ersten MCB
00h	Doppelwort	Zeiger auf den ersten Drive-Parameter-Block
04h	Doppelwort	Zeiger auf den letzten DOS-Buffer-Block
08h	Doppelwort	Zeiger auf den Uhr-Treiber
0Ch	Doppelwort	Zeiger auf den CON-Treiber
10h	Wort	Maximallänge eines Sektors
12h	Doppelwort	Zeiger auf den ersten DOS-Buffer-Block
16h	Doppelwort	Zeiger auf die Pfad-Tabelle
1Ah	Doppelwort	Zeiger auf die System-File-Table

- Im Offset **12h** des **DIB** steht ein Zeiger auf den ersten DOS-Puffer-Block für Plattenzugriffe. Jeder Pufferblock kann 512 Byte aufnehmen. Sein Aufbau ist nachfolgend dargestellt.

Aufbau einer DOS-Puffer-Blockes:

Offset:	Datentyp:	Inhalt:
00h	Doppelwort	Zeiger auf nächsten Puffer-Block
04h	Byte	Logische Nummer des Laufwerkes
05h	Byte	Status-Information
06h	Wort	Sektor-Nummer
08h	Wort	Reserviert
0Ah	512 Bytes	Gepufferter Sektor

45 Funktion 56h: Datei umbenennen oder verschieben

Eingabe: **AH**=56h
DS:DX=Alter Dateiname
ES:DI=Neuer Dateiname

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**

- Die Dateinamen müssen als ASCIIZ-String[6] vorliegen.
- Die Dateinamen können auch Pfad- und Laufwerksnamen beinhalten. Dadurch kann mit dieser Funktion eine Datei auch in ein anderes Verzeichnis verschoben werden.
- Verzeichnissnamen, versteckte Dateien und Systemdateien können nicht geändert bzw. verschoben werden.
- mögliche Fehler:

  02h: Datei nicht gefunden

  03h: Pfad nicht gefunden

  05h: Zugriff verweigert

  11h: Nicht der gleiche Blocktreiber

  50h: Datei mit diesem Namen existiert schon

[6] Ein ASCIIZ-String ist ein String, der mit einem Byte mit der Codierung 00h als „Stringendemarke“ abgeschlossen ist.

46 Funktion 57h: Datum und Zeit einer Datei ermitteln oder setzen

Eingabe: **AH=57h**
AL=Modus (Setzen oder Lesen)
BX=Handle-Nummer
CX=Zeit
DX=Datum

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**; sonst:
CX=Zeit
DX=Datum

- Der Modus gibt an, ob die Werte gesetzt oder gelesen werden sollen:

  00h: Lesen

  01h: Setzen

- Natürlich werden nur beim **Lesen** Werte in **CX** und **DX** zurückgegeben, genauso müssen diese Register nur beim **Setzen** auf die angegebenen Werte eingestellt werden.

- Die Datei muß zuvor mit einer Handle-orientierten Funktion geöffnet worden sein.

- Format von Datum und Zeit siehe Funktion **4Eh**.

- Mögliche Fehler:

  01h: Ungültige Funktion (**AL** > 1)

  06h: Handle ungültig

47 Funktion 58h, Unterfunktion 00h und 01h: Konzept der Speicherverwaltung lesen oder setzen

Eingabe: **AH**=58h
AL=Modus (Setzen oder Lesen)
BX=Strategie

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**; sonst: **AX**=Strategie

- Bedeutung des Modus-Wertes in **AL**:

  00h: Konzept Lesen

  01h: Konzept Setzen

- Natürlich muss nur beim **Setzen** der Strategie **BX** auf den gewünschten Wert eingestellt werden; ebenso liefert die Funktion nur beim **Lesen** den Strategiecode in **AX** zurück.

- DOS verwendet drei verschiedene Strategien zur Zuteilung freien Speichers an einen Prozess entsprechend nachfolgender Aufstellung mit dem jeweiligen zugehörigen Code:

  00h: DOS beginnt die Suche nach einem freien Speicherblock bei den niedrigsten Speicheradressen. Der erste ausreichend große Block wird dem aufrufenden Prozess zugeteilt. Dies nennt man „first fit“.

  01h: DOS durchsucht alle verfügbaren Speicherblöcke und teilt dem aufrufenden Prozess den Speicherblock zu, der den erforderlichen Speicherplatz am geringsten übersteigt. Dieses Verfahren nennt man „best fit“.

  02h: DOS beginnt bei der Suche bei möglichst hohen Speicheradressen. Dieses Verfahren nennt man „last fit“.

  80h: Arbeitet mit 00h, durchsucht jedoch den UMB-Bereich. Wird hier kein Block gefunden, setzt DOS die Suche nach 00h fort.

  81h: Arbeitet mit 01h, durchsucht jedoch den UMB-Bereich. Wird hier kein Block gefunden, setzt DOS die Suche nach 01h fort.

  82h: Arbeitet mit 02h, durchsucht jedoch den UMB-Bereich. Wird hier kein Block gefunden, setzt DOS die Suche nach 02h fort.

  C0h: Arbeitet mit 00h, durchsucht jedoch nur den UMB-Bereich.

  C1h: Arbeitet mit 01h, durchsucht jedoch nur den UMB-Bereich.

  C2h: Arbeitet mit 02h, durchsucht jedoch nur den UMB-Bereich.

- Mögliche Fehler:

  01h: Ungültige Funktion (AL > 3)

  06h: Handle ungültig

  07h: MCB ist zerstört

48 Funktion 58h Unterfunktion 02h: UMB-Verbindungen abfragen

Eingabe: **AH**=58h
AL=02h

Ausgabe: **AL**=UMB-Verbindung

- Die Funktion stellt fest, ob bei der Vergabe von Speicher UMB-Speicher berücksichtigt wird.
- Es bedeutet:

  AL=00h: UMB-Speicherblöcke werden **nicht** eingesetzt.

  AL=01h: UMB-Speicherblöcke finden Verwendung.
- Mögliche Fehler:

  01h: Ungültige Funktion (AL > 3)

  06h: Handle ungültig

  07h: MCB ist zerstört

49 Funktion 58h, Unterfunktion 03h: UMB-Verbindungen setzen

Eingabe: **AH**=58h
AL=03h
BX=UMB-Verbindungen

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**

- Die Funktion legt fest, ob bei der Vergabe von Speicher UMB-Speicher berücksichtigt wird.
- Es bedeutet:

  BX=00h: UMB-Speicherblöcke werden **nicht** eingesetzt.

  BX=01h: UMB-Speicherblöcke finden Verwendung.
- Mögliche Fehler:

  01h: Ungültige Funktion (AL > 3)

  06h: Handle ungültig

  07h: MCB ist zerstört

50 Funktion 59h: Erweiterten Fehler-Code lesen

Eingabe: **AH**=59h
BX=00h

Ausgabe: **AX**=Erweiterter Fehlercode
BH=Fehlerkategorie
BL=Verfahrensvorschlag
CH=Fehlerquelle

- Die Funktion sollte sofort nach dem Erkennen eines Fehlers aufgerufen werden, da nur über den zuletzt aufgetretenen Fehler Informationen beschafft werden können.

Tabelle der erweiterten Fehlercodes:

Code:	Bedeutung:
00h	Kein Fehler
01h	Unbekannte Funktionsnummer
02h	Datei nicht gefunden
03h	Pfad oder Pfadteil nicht gefunden
04h	Zu viele offene Dateien
05h	Zugriff verweigert
06h	Unbekannte Handle-Nummer
07h	MCB ist zerstört
08h	Speicherplatz reicht nicht aus
09h	Unzulässige Speicheradresse
0Ah	Unzulässiges Environment
0Bh	Falsches Parameterblock-Format
0Ch	Falscher Zugriff-Code
0Dh	Unzulässige Daten
0Eh	(Reserviert)
0Fh	Unbekannter Blocktreiber (Laufwerk)
10h	Das zu löschende Verzeichnis ist das aktuelle
11h	Geräte passen nicht zueinander
12h	Keine weiteren Dateien vorhanden
13h	Diskette ist schreibgeschützt
14h	Unbekannter Treiber
15h	Gerät nicht bereit
16h	Unbekannter Befehl
17h	Falscher CRC
18h	Unzulässige Datenlänge
19h	Suchfehler
1Ah	Unzulässiger Gerätetyp
1Bh	Sektor nicht gefunden
1Ch	Papierende
1Dh	Schreibfehler
1Eh	Lesefehler
1Fh	allgemeiner Fehler
20h	Gesperrte Datei
21h	Gesperrter Datenbereich

Code:	Bedeutung:
22h	Unzulässiger Diskettenwechsel
23h	FCB nicht verfügbar
24h	Sharing-Speicherplatz erschöpft
25h	Codepage nicht ansprechbar
26h	Dateiende aufgetreten
27h	Datenträger voll
32h	Funktion nicht unterstützt
33h	Nicht in Umleitungsliste
34h	Datei existiert bereits
35h	Ungültiger Netzname
36h	Netzwerk belegt
37h	Unzulässiges Gerät
38h	Zu viele Befehle
39h	Hardware-Fehler der Netzwerkkarte
3Ah	Übertragungsfehler im Netz
3Bh	Allgemeiner Netzwerkfehler
3Ch	Fehler in Netzwerkschnittstelle
3Dh	Drucker-Warteschlange belegt
3Eh	Druck-Spooler belegt
3Fh	Druckvorgang abgebrochen
40h	Netzwerkname ist gelöscht
41h	Netzwerkzugriff nicht möglich
42h	Unzulässiger Gerätetyp
43h	Netzname ungültig
44h	Zu viele Netznamen
45h	Session-Liste voll
46h	Sharing nicht möglich
47h	Anforderung nicht angenommen
48h	Geräte-Umleitung nicht möglich
50h	Datei existiert schon
51h	FCB existiert bereits
52h	Verzeichnis kann nicht angelegt werden
53h	Abbruch durch Strg-C oder Pause
54h	Netzwerk-Speicherplatz voll
55h	Laufwerk bereits zugeordnet
56h	Passwort ungültig
57h	Parameter ungültig
58h	Schreibfehler im Netzwerk
5Ah	Netzwerktreiber nicht geladen

Die **Fehlerquelle** erläutert, um welche grundsätzliche Art von Fehler es sich gehandelt haben muss. Nachfolgend eine Tabelle der von DOS erkennbaren Fehlerquellen:

Von DOS erkennbare Fehlerquellen:

Code:	Bedeutung:
01h	unbekannt
02h	Blocktreiber mit wahlfreiem Zugriff (Disketten- und Festplattenlaufwerk)
03h	Netzwerk
04h	RAM
05h	Serieller Einheitentreiber (z.B. Drucker, Modem usw.)

Codes der Fehler-Kategorien:

Code:	Bedeutung:
01h	Kein Platz mehr verfügbar (auf Diskette/Platte, im Speicher oder bei den internen Variablen wie Puffer usw.)
02h	Kein direkter Fehler, sondern ein vorübergehendes Zugriffsverbot (z.B. beim Zugriffsversuch auf geschützte Dateien innerhalb eines Netzwerkes)
03h	Zugriff nicht authorisiert (z.B. bei Netzwerken)
04h	Interner Fehler im Betriebssystem
05h	Hardware-Defekt
06h	Interner Fehler im Betriebssystem
07h	Fehler im Anwender-Programm (z.B. Division durch Null)
08h	Datei oder Gegenstand nicht gefunden (Geräte, Handles, usw.)
09h	Datei oder Gegenstand hat falsches Format
0Ah	Datei oder Gegenstand ist gesperrt
0Bh	Probleme mit Speichermedium (falsche oder defekte Diskette)
0Ch	Datei existiert bereits
0Dh	anderweitige Fehlerklasse
0Eh	Operation unmöglich
0Fh	Geräte-Timeout

Mit dem Verfahrensvorschlag zeigt DOS Möglichkeiten auf, den Fehler geeignet zu verwalten. Nachfolgend eine Tabelle der möglichen Codes für den Verfahrensvorschlag:

Codes für Verfahrensvorschlag:

Code:	Bedeutung:
01h	Vorgang wiederholen und dann den Anwender entscheiden lassen, wie zu verfahren ist. (Abbruch, Wiederholen)
02h	Vorgang nach einer Pause wiederholen
03h	Sofern vom Anwender Daten eingegeben wurden, sollten diese nochmals erfragt werden. Dann sollte der fehlerkritische Vorgang wiederholt werden.
04h	Programm normal beenden
05h	Programm sofort beenden. Das System ist im Moment derart unzuverlässig, dass das Programm ohne das Schließen von Dateien usw. sofort verlassen werden sollte.
06h	Fehler ignorieren
07h	Den Anwender auffordern, die Fehlerursache selbst zu beheben (z.B. durch Diskettenwechsel, Drucker einschalten o.ä.)

51 Funktion 5Bh: Neue Datei erstellen

Eingabe:	**AH**=5Bh **CX**=Attribut-Code **DS:DX**=Dateiname	**Ausgabe:**	Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX** wenn kein Fehler, dann: **AX**=Handle-Nummer

- Im Gegensatz zur Funktion **3Ch** wird mit dieser Funktion nur dann eine Datei angelegt, wenn sie nicht bereits existiert.
- Der String des Dateinamens muss als ASCIIZ-String[7] angelegt sein.
- Mögliche Fehlercodes:

  02h: Datei nicht gefunden

  03h: Pfad oder Teil des Pfades nicht gefunden

  04h: Zu viele geöffnete Dateien

  05h: Zugriff verweigert, z.B. „Diskette schreibgeschützt“

  50h: Datei existiert bereits

Bedeutung, wenn das jeweilige Bit des Datei-Attributes gesetzt ist:

Bit:	Bedeutung:
0	Die Datei kann nur gelesen werden
1	Die Datei ist versteckt
2	Es handelt sich um eine System-Datei
3	Die Datei ist noch nicht archiviert
5-15	Reserviert (immer 0)

[7] Ein ASCIIZ-String ist ein String, der mit einem Byte mit der Codierung 00h als „Stringendemarke“ abgeschlossen ist.

52 Funktion 5Ch, Unterfunktion 00h: Datei sperren

Eingabe: **AH**=5Ch
AL=00h
BX=Handle
CX=High-Wort des Offset
DX=Low-Wort des Offset
SI=High-Wort der Datenbereichlänge
DI=Low-Wort der Datenbereichlänge

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**

- Die Routine sperrt den Bereich einer Datei bei installierter File-Sharing-Software.
- Der in **CX:DX** übergebene Offset gibt den Anfang des gesperrten Bereiches bezogen auf den Dateianfang an, der 32-Bit-Wert in **SI:DI** definiert, wieviele Bytes gesperrt werden sollen.
- Nach dem Sperren einer Datei oder eines Dateibereiches kann kein anderes Programm auf den gesperrten Bereich zugreifen.
- Mit den Werten **CX**=00h, **DX**=00h, **SI**=FFFFh und **DI**=FFFFh wird die Datei komplett gesperrt.
- Jeder gesperrte Bereich **muss über die Unterfunktion 01h wieder freigegeben werden**, bevor die Datei geschlossen oder das Programm beendet wird!
- mögliche Fehler:

  01h: File-Sharing nicht installiert

  06h: Handle existiert nicht

  21h: Datei bereits gesperrt

  24h: Liste für Sperreinträge voll

53 Funktion 5Ch, Unterfunktion 01h: Datei freigeben

Eingabe: **AH**=5Ch
AL=01h
BX=Handle
CX=High-Wort des Offset
DX=Low-Wort des Offset
SI=High-Wort der Datenbereichlänge
DI=Low-Wort der Datenbereichlänge

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**

- Die Routine gibt den Bereich einer Datei bei installierter File-Sharing-Software wieder frei, der zuvor mit der Unterfunktion **00h** gesperrt wurde.
- Der in **CX:DX** und in **SI:DI** übergebene Wert muss mit dem Wert übereinstimmen, mit dem zuvor gesperrt wurde. Der entsperrte Bereich kann jedoch auch kleiner sein, dann bleibt der restliche Bereich gesperrt.
- mögliche Fehler:

  01h: File-Sharing nicht installiert

  06h: Handle existiert nicht

  21h: Bereich nicht deckungsgleich mit gesperrtem Bereich

54 Funktion 5Dh, Unterfunktion 0Ah: Erweiterten Fehler setzen

Eingabe: **AH**=5Dh
AL=0Ah
DS:DX=Adresse des Parameterblockes

Ausgabe: keine Ausgabe

- Im Parameterblock stehen die Werte, die die Funktion **59h** zurückliefern soll.

Aufbau des Parameterblockes:

Offset:	Datentyp:	Bedeutung:
00h	Wort	Wert für **AX**
02h	Wort	Wert für **BX**
04h	Wort	Wert für **CX**
06h	Wort	Wert für **DX**
08h	Wort	(Reserviert)
0Ah	Wort	Wert für **DI**
0Ch	Wort	(Reserviert)
0Eh	Wort	Wert für **ES**

55 Funktion 62h: PSP-Adresse ermitteln

Eingabe: **AH**=62h **Ausgabe:** **BX**=Segment-Adresse des PSP

- Bei **COM**-Dateien wird der PSP ab dem aktuellen Codesegment abgelegt, bei **EXE**-Dateien kann er aufgrund von segmentübergreifendem Code auch woanders liegen.
- Die Adresse bezieht sich immer auf den PSP des aktuellen Programms, nicht etwa auf einen mit Funktion **26h** erstellten PSP eines Tochterprogrammes. (Siehe auch Funktion **51h**)

56 Funktion 67h: Maximale Handle-Zahl setzen

Eingabe: **AH**=67h
BX=Handle-Anzahl
Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**

- Wird im **BX**-Register ein Wert von weniger als 20 angegeben, so setzt DOS die Handle-Anzahl aud 20.
- Wird die Handle-Anzahl auf mehr als **FFh** gesetzt, benötigt das Betriebssystem mehr Platz in der Handle-Liste. In diesem Fall muß die Größe des aktuellen Speicherbereiches erhöht werden.
- Mögliche Fehler:

  04h: zu viele offene Dateien

  08h: Speicherplatz reicht nicht aus

57 Funktion 68h: Dateipuffer schreiben

Eingabe: **AH**=68h
BX=Handle-Nummer
Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**

- Diese Funktion sorgt dafür, dass beim Speichern von Dateien keine Daten verlorengehen können. Vor allem bei der Arbeit mit CACHE-Speichern oder großen Disketten-(Festplatten-)Puffern ist nicht immer sicher, ob Daten, die an die Diskette (Festplatte) geschickt werden, tatsächlich schon physikalisch auf ihr gespeichert sind oder noch in einem Puffer stehen.
- Die Datei wird bei dieser Funktion nicht geschlossen.
- Die Funktion ersetzt die sonst mögliche Sequenz schließen/öffnen, die früher häufig für den gleichen Zweck verwendet wurde, ist aber erheblich schneller.
- Möglicher Fehler:

  06h: unbekannte Handle-Nummer

58 Funktion 6Ch: Erweitertes Öffnen einer Datei

Eingabe: **AH**=6Ch
BX=Modus
CX=Datei-Attribut
DX=Reaktion
DS:SI=Adresse des Dateinamens

Ausgabe: Carry-Flag zeigt Fehler an, dann: Fehlercode in **AX**, sonst: **AX**=Handle-Nummer

- Der Dateiname muß als ASCIIZ-String[8] vorliegen.
- Zur Bedeutung des Datei-Attributes vergleiche Funktion **43h**.
- Der Modus-Parameter legt die Art des Zugriffes fest (siehe nachfolgende Tabelle).

Modus-Parameter in BX:

Bit:	Wert:	Bedeutung:
15	**0**	Immer 0
14	**1**	Der Dateieintrag wird nach jedem Schreibzugriff aktualisiert.
	0	Der Dateieintrag wird nicht beständig aktualisiert.
13	**1**	Beim Auftreten eines Fehlers wird Interrupt **24h** ausgelöst.
	0	Interrupt **24h** wird nicht ausgelöst, sondern der Fehlercode an das aufrufende Programm übergeben.
12-8	**0**	(Reserviert)
7	**1**	Beim Starten eines Tochterprogrammes werden die Zugriffsrechte nicht vererbt.
	0	Die Zugriffsrechte werden vererbt.
6-4	**000b**	Die Datei wird nur für ein Programm geöffnet (Kompatibilitätsmodus).
	001b	Die Datei wird nur für ein Programm geöffnet.
	010b	Die Datei wird für das aktuelle Programm zum Lesen und Schreiben geöffnet, andere Prozesse dürfen sie nur lesen.
	011b	Die Datei wird für das aktuelle Programm zum Lesen und Schreiben geöffnet, andere Prozesse dürfen sie nur schreiben.
	100b	Die Datei wird für das aktuelle Programm und auch andere Prozesse zum Lesen und Schreiben geöffnet.
3-2	**0**	(Reserviert)
1-0	**00b**	Die Datei darf nur gelesen werden.
	01b	Die Datei darf nur geschrieben werden.
	10b	Die Datei darf gelesen und geschrieben werden.

- Mit dem Reaktion-Parameter im DX-Register wird festgelegt, wie DOS reagieren soll, wenn die Datei bereits existiert. Nachfolgend die Tabelle mit den zulässigen Werten.

[8] Ein ASCIIZ-String ist ein String, der mit einem Byte mit der Codierung 00h als „Stringendemarke“ abgeschlossen ist.

Reaktions-Parameter in BX:

Code:	Bedeutung:
0000h	Falls die Datei nicht existiert, mit einer Fehlermeldung abbrechen, ansonsten die Datei zum Lesen öffnen.
0001h	Falls die Datei nicht existiert, mit einer Fehlermeldung abbrechen, ansonsten die Datei überschreiben.
0002h	Falls die Datei nicht existiert, Erstellen der Datei zum Lesen und Schreiben, ansonsten mit einer Fehlermeldung abbrechen.

- Mögliche Fehler:

  01h: Unbekannte Funktionsnummer

  02h: Datei nicht gefunden

  03h: Pfad nicht gefunden

  04h: Zu viele offene Dateien

  05h: Zugriff verweigert

  0Ch: Unzulässige Daten

  20h: Gesperrte Datei

  20h: Gesperrte Datei

  50h: Datei existiert schon

59 Erklärung einiger wichtiger Begriffe

PSP = Programm-Segment-Präfix: In diesem 256-Byte großen Bereich verwaltet DOS bei jedem Programm die programmspezifischen Daten. Bei COM-Programmen beginnt er im Codesegment an der Adresse 0000h, das eigentliche Programm beginnt dann bei 0100h. Bei EXE-Programmen liegt der Bereich in einem anderen Segment, das mit Funktion **51h** oder **62h** ermittelt werden kann.

DTA = Data-Transfer-Area: Dieser Bereich ab Offset **80h** bis **FFh** im PSP wird für einige Funktionen zum Datenaustausch mit dem Betriebssystem verwendt, z.B. für den Übergabestring oder bei der Dateisuche.

MCB = Memory-Controll-Block: Dieser 16-Byte-große Block liegt immer direkt vor jedem belegten oder freien Speicherblock im RAM. Darin stehen Informationen über die Verwendung und die Größe des nachfolgenden Speicherblockes.

Handle: Alle Zugriffe auf (geöffnete) Dateien erfolgen stets über eine Nummer im Wortformat als Identifizierungsmerkmal. Diese Nummer nennt man „File-Handle“ oder auf Deutsch „Datei-Handle“ oder eben nur kurz „Handle“. Lediglich zum Öffnen oder Anlegen einer Datei ist der komplette Dateiname notwendig. Beim Öffnen oder Anlegen teilt DOS dem aufrufenden Programm dieses „Handle“ mit. Das Programm muss sich die Nummer merken, bis die Datei wieder geschlossen ist.