

FC.BAS	0.02	25. 6.2025
--------	------	------------

**B E S C H R E I B U N G :**

- Ersetzt FC.EXE und vergleicht Dateien.

**B E A R B E I T U N G :**

21. 9.2024 - 25. 6.2025      Norbert Südland, Aalen

**VORBEREITUNG:**

'OPTION EXPLICIT      'Nützlich bei Visual Basic

CONST Inhalt1\$ = "Inhalt\_1. \$\$\$"

CONST Inhalt2\$ = "Inhalt\_2. \$\$\$"

DECLARE FUNCTION LASTINSTR% (Text\$, Suchausdruck\$)

DECLARE SUB LINEINPUT (Datei%, Zeile\$)

DECLARE SUB Pause ()

DEF FNMIN (a, b) = (a + b) / 2 - ABS(a - b) / 2

DEF FNMAX (a, b) = (a + b) / 2 + ABS(a - b) / 2

DIM Argument1\$, Argument2\$

DIM Ausgabertext\$

DIM Befehl\$

DIM Datensatz1\$, Datensatz2\$

DIM Pfad\$, Pfad1\$, Pfad2\$

DIM Zeile\$, Zeile1\$, Zeile2\$

DIM z%

DIM Datei1%, Datei2%, Datei3%, Datei4%

DIM gefunden%

DIM Laenge%

DIM Seiten%

DIM Zeilen%

DIM p1&, p2&

**HAUPTTEIL:**

'Vorbereitung:

'-----'

ON ERROR GOTO Fehler

Seiten% = 0

Zeilen% = 0

```
gefunden% = 0
Argument1$ = ENVIRON$("FC_1")
IF Argument1$ = "" THEN GOTO Programmende
Pfad1$ = LEFT$(Argument1$, LASTINSTR$(Argument1$, "\"))
Befehl$ = "DIR " + Argument1$ + " > " + Inhalt1$
10 SHELL Befehl$

Argument2$ = ENVIRON$("FC_2")
IF Argument2$ = "" THEN GOTO Programmende
Pfad2$ = LEFT$(Argument2$, LASTINSTR$(Argument2$, "\"))
Befehl$ = "DIR " + Argument2$ + " > " + Inhalt2$
20 SHELL Befehl$

'Dateilisten abarbeiten:
'-----'
Datei1% = FREEFILE
30 OPEN Inhalt1$ FOR BINARY ACCESS READ AS #Datei1%
DO
    CALL LINEINPUT(Datei1%, Zeile1$)
    LOOP UNTIL INSTR(2, Zeile1$, ":\")
    Datei2% = FREEFILE
40 OPEN Inhalt2$ FOR BINARY ACCESS READ AS #Datei2%
DO
    CALL LINEINPUT(Datei2%, Zeile2$)
    LOOP UNTIL INSTR(2, Zeile2$, ":\")
DO
    CALL LINEINPUT(Datei1%, Zeile1$)
    IF Zeile1$ = "" THEN
        CALL LINEINPUT(Datei1%, Zeile1$)
    END IF
    IF MID$(Zeile1$, 9, 1) = SPACE$(1) THEN
        Zeile$ = RTRIM$(LEFT$(Zeile1$, 8)) + "."
        Zeile$ = Zeile$ + RTRIM$(MID$(Zeile1$, 10, 3))
        Zeile1$ = Zeile$
    ELSE
        DO
            CALL LINEINPUT(Datei1%, Zeile1$)
            LOOP UNTIL EOF(Datei1%)
        END IF

        CALL LINEINPUT(Datei2%, Zeile2$)
        IF Zeile2$ = "" THEN
            CALL LINEINPUT(Datei2%, Zeile2$)
        END IF
        IF MID$(Zeile2$, 9, 1) = SPACE$(1) THEN
            Zeile$ = RTRIM$(LEFT$(Zeile2$, 8)) + "."
            Zeile$ = Zeile$ + RTRIM$(MID$(Zeile2$, 10, 3))
            Zeile2$ = Zeile$
        ELSE
            DO
                CALL LINEINPUT(Datei2%, Zeile2$)
                LOOP UNTIL EOF(Datei2%)
            END IF

            IF Pfad1$ <> Pfad2$ THEN
                WHILE NOT EOF(Datei1%) AND NOT EOF(Datei2%) AND Zeile1$ < Zeile2$
                    CALL LINEINPUT(Datei1%, Zeile1$)
                WEND
```

```
        WHILE NOT EOF(Datei1%) AND NOT EOF(Datei2%) AND Zeile1$ > Zeile2$
62      CALL LINEINPUT(Datei2%, Zeile2$)
        WEND
        IF Zeile1$ <> Zeile2$ GOTO Schleifenende
    END IF
    IF NOT EOF(Datei1%) AND NOT EOF(Datei2%) THEN
        Ausgabetext$ = "Vergleichen der Dateien " + Pfad1$ + Zeile1$
        Ausgabetext$ = Ausgabetext$ + " und " + Pfad2$ + Zeile2$
        Zeilen% = Zeilen% + LEN(Ausgabetext$) / 80 + 1
        IF INT(Zeilen% / 24) > Seiten% THEN
            Pause
            Seiten% = Seiten% + 1
        END IF
        PRINT Ausgabetext$
        Datei3% = FREEFILE
        Pfad$ = Pfad1$ + Zeile1$
70      OPEN Pfad$ FOR BINARY ACCESS READ SHARED AS #Datei3% LEN = 512
        Datei4% = FREEFILE
        Pfad$ = Pfad2$ + Zeile2$
80      OPEN Pfad$ FOR BINARY ACCESS READ SHARED AS #Datei4% LEN = 512
        Datensatz1$ = SPACE$(512)
        Datensatz2$ = SPACE$(512)
        p1& = 1
        p2& = 1
        WHILE NOT EOF(Datei3%) AND NOT EOF(Datei4%) AND gefunden% = 0
90      GET #Datei3%, p1&, Datensatz1$
100     GET #Datei4%, p2&, Datensatz2$
        IF Datensatz1$ = Datensatz2$ THEN
            p1& = p1& + 512
            p2& = p2& + 512
            IF gefunden% = 0 THEN
                Ausgabetext$ = "FC: Keine Unterschiede gefunden"
            ELSE
                Ausgabetext$ = ""
            END IF
        ELSE
            GOSUB Unterschied1
            Ausgabetext$ = ""
        END IF
        WEND
        Zeilen% = Zeilen% + INT((LEN(Ausgabetext$) - 1) / 80) + 1
        IF INT(Zeilen% / 24) > Seiten% THEN
            Pause
            Seiten% = Seiten% + 1
        END IF
        IF Ausgabetext$ <> "" THEN
            PRINT Ausgabetext$
            Zeilen% = Zeilen% + 1
            IF INT(Zeilen% / 24) > Seiten% THEN
                CALL Pause
                Seiten% = Seiten% + 1
            END IF
            PRINT
        END IF
        CLOSE Datei4%
        CLOSE Datei3%
    END IF
'-----'
```

Schleifenende:

```
'-----'  
        LOOP WHILE NOT EOF(Datei2%) AND NOT EOF(Datei1%)  
        CALL Pause  
        'STOP  
        CLOSE Datei2%  
        CLOSE Datei1%
```

110 KILL Inhalt2\$

120 KILL Inhalt1\$

'====='

Programmende:

'====='

SYSTEM

' \_\_\_\_\_ ENDE DES HAUPTTEILS \_\_\_\_\_ '

FEHLERBEHANDLUNG:

'====='

Fehler:

'====='

PRINT "ERR = "; ERR, "ERL = "; ERL

STOP

ON ERROR GOTO 0

GOTO Programmende

' \_\_\_\_\_ ENDE DER FEHLERBEHANDLUNG \_\_\_\_\_ '

UNTERPROGRAMM UND FUNKTION:

'-----'

Unterschied1:

'-----'

200 PRINT "FC: Dateien sind unterschiedlich:"

Laenge% = FNMIN(LEN(Datensatz1\$), LEN(Datensatz2\$))

FOR z% = 1 TO Laenge%

IF MID\$(Datensatz1\$, z%, 1) <> MID\$(Datensatz2\$, z%, 1) THEN

gefunden% = z%

z% = Laenge%

END IF

NEXT

Laenge% = FNMIN(Laenge% - gefunden%, 79)

Ausgabertext\$ = MID\$(Datensatz1\$, gefunden%, Laenge%)

COLOR 0, 7

PRINT Ausgabertext\$;

COLOR 7, 0

PRINT

PRINT " und"

Ausgabertext\$ = MID\$(Datensatz2\$, gefunden%, Laenge%)

COLOR 0, 7

PRINT Ausgabertext\$;

COLOR 7, 0

```
PRINT
RETURN 'Unterschied1 _____'

'=====
FUNCTION LASTINSTR% (Text$, Suchausdruck$)
'=====
' Ermittelt das letzte Auftreten von 'Suchausdruck$' in 'Text$'.
'
' Bearbeitung:
' 21. 9.2024 Norbert Südland, D-73431 Aalen
'-----

DIM Beginn%
DIM LetzterBeginn%

Beginn% = 0
DO
    LetzterBeginn% = Beginn%
    Beginn% = INSTR(Beginn% + 1, Text$, Suchausdruck$)
LOOP WHILE Beginn% > 0
LASTINSTR = LetzterBeginn%
END FUNCTION 'LASTINSTR _____

'=====
SUB LINEINPUT (Datei%, Zeile$)
'=====
' Ersetzt den Befehl LINE INPUT, um auch unter der DOS-Box 0.73 zu laufen.
' Diese DOS-Box erzeugt mit DIR eine Ausgabe wie unter Unix und kann mit
' LINEINPUT statt mit LINE INPUT ausgewertet werden.
' Die 'Datei%' muss dazu als BINARY ACCESS READ (WRITE) geöffnet sein.
' Am Ende wird eine Zeile mit 512 Zeichen zurück gegeben, damit jede Zeile,
' die nicht mit CHR$(10) endet, noch erfasst wird. Dieser Kompromiss sorgt
' für Geschwindigkeit und reicht hier für das Auswerten von DIR aus.
'
' Bearbeitung:
' 11.08.2009: Norbert Südland, 73431 Aalen, GERMANY
' Überprüfung:
' 13.08.2009: Norbert Südland, 73431 Aalen, GERMANY
'-----

DIM AltePosition&
DIM Laenge&
DIM NeuePosition&

Zeile$ = SPACE$(512)
AltePosition& = SEEK(Datei%)
IF EOF(Datei%) THEN
    Zeile$ = ""
    GOTO Ende
END IF

GET #Datei%, AltePosition&, Zeile$
Laenge& = INSTR(Zeile$, CHR$(10))
IF Laenge& <> 0 THEN
    Zeile$ = LEFT$(Zeile$, Laenge& - 1)
    IF INSTR(RIGHT$(Zeile$, 1), CHR$(13)) <> 0 THEN
        Zeile$ = LEFT$(Zeile$, LEN(Zeile$) - 1)
    END IF
END IF
END IF
```

```
IF EOF(Datei%) = 0 OR Laenge& <> 0 THEN
    NeuePosition& = AltePosition& + Laenge&
    SEEK #Datei%, NeuePosition&
END IF
'---'
Ende:
'---'
END SUB 'LINEINPUT _____'

'====='
SUB Pause
'====='
' Macht eine Pause, wenn der Bildschirm vollgeschrieben ist.
'
' Bearbeitung:
' 18.10.2024 - 25.06.2025 Norbert Südland, Aalen
'-----'
DIM x%      'AS INTEGER
DIM y%      'AS INTEGER

'Aktuelle Cursor-Position ermitteln:
'-----'
x% = CSRLIN
y% = POS(0)

'Zeile 25 löschen und beschriften:
'-----'
LOCATE 25, 1, 0
PRINT SPACE$(80);
LOCATE 25, 1, 1
PRINT " -- Fortsetzung  -- ";

'Tastendruck abwarten:
'-----'
WHILE LEN(INKEY$) = 0
WEND

'Zeile 25 erneut löschen:
'-----'
LOCATE 25, 1, 1
PRINT SPACE$(80);

'Cursor-Position restaurieren:
'-----'
LOCATE x%, y%
END SUB 'Pause _____'
```