```
LOOP2:  MOV     AL,TIEFE_WORK
        XOR     AH,AH
        MOV     DI,AX
        MOV     BL,TIEFE_STATUS[DI]     ;BL:=TIEFE_STATUS[TIEFE_WORK]
        CMP     AL,TIEFE_AKTUELL        ;falls TIEFE_WORK < TIEFE_AKTUELL,
        JNB     AKT_NAME                ;dann nur Ausgabe von entspr. Linien.
        CMP     BL,1                    ;folgen in dieser Spalte
        MOV     SI,OFFSET LINE0         ;noch Namen?
        JNE     NO_LINE                 ;wenn ja,
        MOV     SI,OFFSET LINE1         ;dann Ausgabe von |
NO_LINE:
        MOV     CX,2
        CALL    STRING_OUT              ;Ausgabe von Leerzeichen oder |
        MOV     SI,OFFSET BLANK_ROW
        MOV     CX,NLEN
        CALL    STRING_OUT              ;Ausgabe von Leerzeichen
        INC     TIEFE_WORK
        JMP     SHORT LOOP2

;Nächster/erster Name suchen und ausgeben:

AKT_NAME:
        CMP     BL,1                    ;schon ein Name ausgegeben?
        JNE     RD_FIRST
        MOV     FIRST,0                 ;für später
COPY1:  CALL    COPY                    ;zuletzt gelesener Name sichern
        MOV     AH,NEXT_ENTRY
        CALL    READ                    ;nächster Name lesen
        MOV     AL,FIRST
        JC      STATUS2                 ;falls letzter Name -> STATUS2
STATUS1:
        MOV     AH,1                    ;Status := 1 (es folgen noch Namen)
        AND     AL,AL
        MOV     SI,OFFSET LINE4
        JZ      LIN_OUT                 ;dies ist nicht der 1. N., also ├
        MOV     SI,OFFSET LINE5         ;dies ist der erste Name, also ─┬
        JNZ     LIN_OUT
STATUS2:
        MOV     AH,2                    ;Status := 2 (es ist der letzte Name)
        AND     AL,AL
        MOV     SI,OFFSET LINE2
        JNZ     LIN_OUT                 ;es ist der einzigste Name, also ──
        MOV     SI,OFFSET LINE3         ;es ist der letzte von mehreren, └
LIN_OUT:
        MOV     BL,TIEFE_AKTUELL
        MOV     BH,0
        MOV     DI,BX
        MOV     TIEFE_STATUS[DI],AH     ;TIEFE_STATUS[TIEFE_AKTUELL]:=AH
        MOV     CX,2
        CALL    STRING_OUT              ;Ausgabe von ──, └, ├, oder ─┬
        CALL    AKTUELL_OUT             ;Ausgabe des gesicherten Namens
        INC     TIEFE_AKTUELL           ;nachfolgendes Directory durchsuchen

RD_FIRST:
        MOV     AH,FIRST_ENTRY          ;erster Name des Directory's suchen
        CALL    READ
        JC      NO_NAME                 ;kein Name, also zurück
        MOV     FIRST,1
        MOV     AL,TIEFE_AKTUELL
        CMP     AL,TIEFE_MAX            ;falls TIEFE_MAX nicht überschritten
        JNA     COPY1                   ;dann Ausgabe des neuen Namens
        MOV     DL,196                  ; '─'
        MOV     AH,OUT_CON              ;ansonsten nur andeuten, daß weitere
        INT     DOS                     ;Namen folgen

NO_NAME:
        DEC     TIEFE_AKTUELL           ;eine Verzweigungstiefe zurück
        CMP     TIEFE_AKTUELL,0         ;falls Hauptdirectory überschritten,
        JL      BAUM_END                ;dann Kommando beenden
        MOV     AL,TIEFE_AKTUELL
        XOR     AH,AH
        MOV     DI,AX                   ;falls dieses Directory auch
        CMP     TIEFE_STATUS[DI],2      ;vollständig ausgegeben wurde,
        JE      NO_NAME                 ;dann nochmals eins zurück
        JMP     LOOP1

BAUM_END:
        CALL    LFCR_OUT
        XOR     AL,AL                   ;kein Fehler aufgetreten
END2:   MOV     AH,TERMINATE            ;Kommando beenden
        INT     DOS

DTA_ADRESS      DW      ?

READ            PROC    NEAR

;Diese Prozedur sucht nach dem ersten oder nächsten Directory-Namen
;(AH=4EH oder 4FH).
;Es kopiert abhängig von TIEFE_AKTUELL den neuen Pfadnamen aus DIR_NAMES nach
;FILE_NAME. (DIR_NAMES wird von Prozedur COPY ständig aktualisiert.)
;CF=1 falls es keine Namen mehr gibt, ansonsten kann mit COPY und
;AKTUELL_OUT der gefundene Name ausgeben werden.

        PUSH    AX                      ;AH für INT DOS retten

        MOV     AL,2BH
        MUL     TIEFE_AKTUELL
        ADD     AX,OFFSET DTA_BUF       ;AX:= DTA_BUF+TIEFE_AKTUELL*2BH
        MOV     DX,AX                   ;DX:= AX=aktueller DTA-Buffer
        MOV     DTA_ADRESS,AX
        MOV     AH,SET_DTA              ;DTA-Buffer-Adresse setzen
        INT     DOS

        MOV     AL,NLEN+1
        MUL     TIEFE_AKTUELL
        CLD
        MOV     CX,AX                   ;CX:=Länge des neuen Pfades
        LEA     SI,DIR_NAMES
        MOV     DI,FILE_BEG
        AND     CX,CX
        JZ      NO_COPY
        REP     MOVSB                   ;neuer Pfad nach FILE_NAME kopieren
NO_COPY:
        MOV     AL,'\'
        STOSB
        LEA     SI,STARS
        MOV     CX,4
        REP     MOVSB                   ;\*.* nach FILE_NAME kopieren

        POP     AX                      ;AH wieder herstellen
READ_LP:
        MOV     CX,ATTRIB               ;Attribut für Directory-Name
        LEA     DX,FILE_NAME
        INT     DOS                     ;Name suchen
        JC      READ_END                ;keiner gefunden?
```