

```

MOV     BX,DTA_ADDRESS
MOV     AL,CBXJ+15H
MOV     AH,NEXT_ENTRY
AND     AL,ATTRIB
JZ      READ_LP
MOV     AL,CBXJ+1EH
CMP     AL,'.'
JE      READ_LP
CLC
READ_END: RET

READ     DB      LF,CR
LFCR_OUT PROC    NEAR
;Diese Prozedur platziert den Cursor an den Anfang der nächsten Zeile
MOV     SI,OFFSET LFCR
MOV     CX,2
CALL   STRING_OUT
RET

LFCR_OUT ENDP
PROC    NEAR
;Diese Prozedur kopiert vom aktuellen DTA-Buffer den Directory-Namen mit einem
; \ voran gestellt in die entsprechende Position von DIR_NAMES.
; DIR_NAMES wird von READ und AKTUELL_OUT benutzt.
MOV     AL,2BH
MUL     TIEFE_AKTUELL
MOV     SI,AX
SI,OFFSET DTA_BUF+1EH
MOV     AL,NLEN+1
MUL     TIEFE_AKTUELL
ADD     AX,OFFSET DIR_NAMES
MOV     DI,AX
MOV     AL,' \ '
CX,NLEN
CALL   COPY_LP
AND     AL,AL
AND     COPY_ZERO
JZ      STOSB
DEC     CX
AND     COPY_LP
RET

COPY_ZERO: MOV     AL,' '
REP     STOSB
RET

COPY     ACTUELL_OUT PROC    NEAR
; Diese Prozedur gibt den aktuellen Directory-Namen aus.
; Er wird aus DIR_NAMES abhängig von TIEFE_AKTUELL kopiert.
STD
MOV     AL,NLEN+1
MUL     TIEFE_AKTUELL
MOV     BX,AX
MOV     AX,OFFSET DIR_NAMES+NLEN
MOV     SI,AX
CX,NLEN
CALL   AKT_LP
MUL     TIEFE_AKTUELL
MOV     AL,' '
MOV     AKT_OUT
CX
DL,196
MOV     AH,OUT_CON
DOS
SHORT AKT_LP
MOV     BX,OFFSET DIR_NAMES+1
MOV     SI,BX
CALL   STRING_OUT
RET
; Ausgabe des Namens
ACTUELL_OUT ENDP
PROC    NEAR
; Diese Prozedur gibt ab [SI] CX Zeichen aus
STRING_OUT PROC    NEAR
MOV     AH,OUT_CON
DL,AL
DOS
CX
AND     CX,CX
JNZ    STRING_LP
RET
STRING_OUT ENDP
CODE   ENDS
END

```

```

;AL:= tatsächliches Attribut
;ist es der Name eines Directory's?
;ist es . oder ..?

BX,DTA_ADDRESS
AL,CBXJ+15H
AH,NEXT_ENTRY
AL,ATTRIB
JZ,READ_LP
MOV,AL,CBXJ+1EH
CMP,AL,'.'
JE,READ_LP
CLC
READ_END: RET

READ DB LF,CR
LFCR_OUT PROC NEAR
;Diese Prozedur platziert den Cursor an den Anfang der nächsten Zeile
MOV SI,OFFSET LFCR
MOV CX,2
CALL STRING_OUT
RET

LFCR_OUT ENDP
PROC NEAR
;Diese Prozedur kopiert vom aktuellen DTA-Buffer den Directory-Namen mit einem
; \ voran gestellt in die entsprechende Position von DIR_NAMES.
; DIR_NAMES wird von READ und AKTUELL_OUT benutzt.
MOV AL,2BH
MUL TIEFE_AKTUELL
MOV SI,AX
SI,OFFSET DTA_BUF+1EH
MOV AL,NLEN+1
MUL TIEFE_AKTUELL
ADD AX,OFFSET DIR_NAMES
MOV DI,AX
MOV AL,' \ '
CX,NLEN
CALL COPY_LP
AND AL,AL
AND COPY_ZERO
JZ STOSB
DEC CX
AND COPY_LP
RET

COPY_ZERO: MOV AL,' '
REP STOSB
RET

COPY ACTUELL_OUT PROC NEAR
; Diese Prozedur gibt den aktuellen Directory-Namen aus.
; Er wird aus DIR_NAMES abhängig von TIEFE_AKTUELL kopiert.
STD
MOV AL,NLEN+1
MUL TIEFE_AKTUELL
MOV BX,AX
MOV AX,OFFSET DIR_NAMES+NLEN
MOV SI,AX
CX,NLEN
CALL AKT_LP
MUL TIEFE_AKTUELL
MOV AL,' '
MOV AKT_OUT
CX
DL,196
MOV AH,OUT_CON
DOS
SHORT AKT_LP
MOV BX,OFFSET DIR_NAMES+1
MOV SI,BX
CALL STRING_OUT
RET
; Ausgabe des Namens
ACTUELL_OUT ENDP
PROC NEAR
; Diese Prozedur gibt ab [SI] CX Zeichen aus
STRING_OUT PROC NEAR
MOV AH,OUT_CON
DL,AL
DOS
CX
AND CX,CX
JNZ STRING_LP
RET
STRING_OUT ENDP
CODE ENDS
END

```

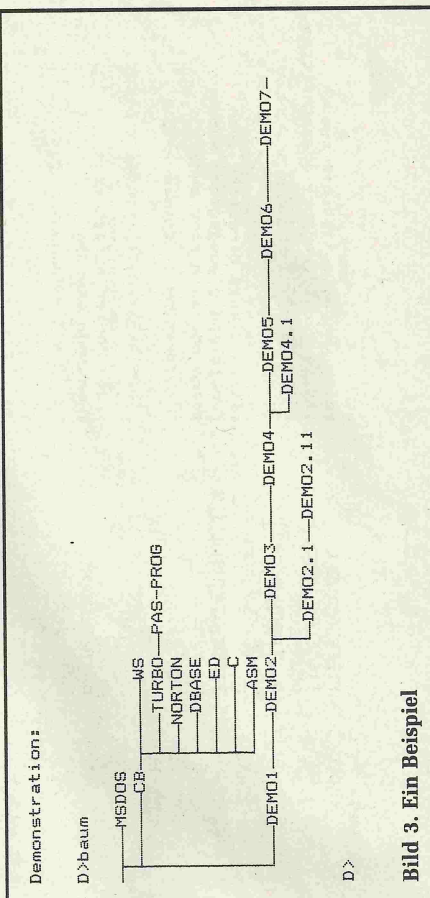


Bild 3. Ein Beispiel