

```

;*****;
;*          P R C V T          *;
;*-----*
;* Task      : Points the BIOS printer interrupt to its own *;
;*            routine and makes it possible for example *;
;*            to convert IBM-ASCII to EPSON. *;
;*            The program is de-installed on the *;
;*            second call and removed from memory. *;
;*-----*
;* Author    : Michael Tischer *;
;* Developed on : 08/02/87 *;
;* Last update : 02/14/92 *;
;*-----*
;* Assembly  : MASM PRCVT; *;
;*            LINK PRCVT; *;
;*            EXE2BIN PRCVT PRCVT.COM or *;
;*            *;
;*            TASM PRCVT *;
;*            TLINK PRCVT /T *;
;*-----*
;* Call      : PRCVT *;
;*****;

```

== Actual program starts here ==

```

code      segment para 'CODE'      ;Definition of the CODE segment

org 100h

assume cs:code, ds:code, es:code, ss:code

start:    jmp prcvtini             ;The first executable command

== Data (remain in memory) ==

olderint  equ this dword           ;Old interrupt vector 17H
intoldofs dw (?)                   ;Offset address Interrupt vector 17H
intoldseg dw (?)                   ;Segment address Interrupt vector 17H

;-- The following table contains the new -----
;-- code followed by the old code -----

codetab   db 64, 21                ; Paragraph --- > '@'
          db 47,201                ; 'É' -----> '/'
          db 124,186               ; '°' -----> '|'
          db 92,200                ; 'È' -----> '\'
          db 45,205                ; 'Í' -----> '-'
          db 92,187                ; '»' -----> '\'
          db 47,188                ; '¼' -----> '/'
          db 43,206                ; 'Î' -----> '+'
          db 0                    ;End of the table

```

== this is the new printer interrupt (remains in memory) ==

```

newpri    proc far

          jmp short newpri_1

          db "CW"                  ;Identification of the program

newpri_1: or ah,ah                 ;Print character (function 0)?
          jne aint                 ;No --> Call old interrupt

          pushf                    ;All registers changed in the
          push bx                   ;program must be stored
          push si
          push ds

          push cs                   ;Save CS on the stack
          pop ds                   ;Get DS from stack

;-- Does code have to be converted ? -----

          cld                      ;Increment on string instructions
          mov si,offset codetab     ;Code table address
          mov bl,al                ;Store code in BL

```

```

testcode: lodsw                ;Load old (AH) and new code (AL)
          or  al,al            ;Reached end of table?
          je  notfound        ;Yes --> Code not found
          cmp ah,bl           ;Is it the code for conversion?
          jne testcode        ;No --> Continue to search table
          jmp short nreset     ;It was a code for conversion

notfound: mov  al,bl           ;Move old code to AL again
nreset:   xor  ah,ah          ;Set function number 0 again
          pop  ds             ;Restore registers
          pop  si
          pop  bx
          popf

aint:    jmp  cs:[olderint]   ;Go to old printer routine

newpri    endp

instend   equ this byte      ;Everything must remain resident
                               ;up to this memory location

;== Data (can be overwritten by DOS) =====
installm  db 13,10,"PRCVT (c) 1987,92 by Michael Tischer",13,10,13,10
          db "PRCVT installed. Call PRCVT again to de-install.",13,10,"$"

removeit  db 13,10,"PRCVT de-installed.",13,10,"$"

;== Program (can be overwritten by DOS) =====
;-- Start and initialization routine -----
prcvtini  label near

          mov ax,3517h        ;Get contents of interrupt vector 17H
          int 21h             ;Call DOS interrupt
          cmp word ptr es:[bx+2],"WC" ;Test if PRCVT exists
          jne install        ;Not installed --> INSTALL

          ;-- PRCVT was de-installed -----

          mov dx,es:intoldofs  ;Offset address of interrupt 17H
          mov ax,es:intoldseg  ;Segment address of interrupt 17H
          mov ds,ax           ;to DS
          mov ax,2517h        ;Redirect interrupt control
          int 21h             ;Vector 17H to old routine

          mov bx,es
          mov es,es:[2Ch]

          mov ah,49h          ;Release storage of old PRCVT
          int 21h             ;Call DOS interrupt

          mov es,bx
          mov ah,49h
          int 21h
          mov ah,49h
          int 21h

          push cs              ;Store CS on stack
          pop  ds              ;Restore DS

          mov dx,offset removeit ;Message: Program removed
          mov ah,9             ;Write function number for string
          int 21h              ;Call DOS interrupt

          mov ax,4C00h         ;End program
          int 21h              ;Call DOS interrupt to end

          ;-- install PRCVT -----

install   label near

          mov ax,3517h        ;Get contents of interrupt vector 17H
          int 21h             ;Call DOS interrupt

```

```

mov  intoldseg,es      ;Save segment and offset addresses
mov  intoldofs,bx      ;of interrupt vector 17H

mov  dx,offset newpri   ;Offset address new interrupt routine
mov  ax,2517h          ;Redirect contents of interrupt
int  21h               ;vector 17H to user routine

mov  dx,offset installm ;Message: Program installed
mov  ah,9              ;Function number: Display string
int  21h               ;Call DOS interrupt

;-- Only the PSP, the new interrupt routine and the -----
;-- data pertaining to it must remain resident. -----

mov  dx,offset instend  ;Calculate the number of 16-byte
mov  cl,4               ;paragraphs available
shr  dx,cl              ;to the program
inc  dx
mov  ax,3100h           ;End program with end code 0 (0.0)
int  21h               ;but remain resident

;== End =====

code    ends            ;End of CODE segment
end     start

```