

Pascal listing: RAWCOOK.PAS

```
{*****}
{*               R A W C O O K               *}
{*-----*}
{* Task          :   provide two functions to switch   *}
{*               :   a character device driver to the RAW- *}
{*               :   or the COOKED mode                 *}
{*-----*}
{* Author        :   MICHAEL TISCHER                   *}
{* developed     :   08/16/87                           *}
{* last Update   :   04/07/95                           *}
{*****}
```

program RAWCOOKP;

Uses Crt, Dos; { CRT and DOS units }

const STANDARDIN = 0; { handle 0 is connected with Standard input }
STANDARDOUT = 1; { handle 1 is connected with Standard output }

var Keys : char; { only needed for Demo program }

```
{*****}
{* GETMODE: read attribute of device driver in          *}
{* Input   : the handle passed must be connected to device addressed *}
{* Output  : the device attribute                        *}
{*****}
```

function GetMode(Handle : integer) : integer;

var Regs : Registers; { register-Variable for Interrupt call }

begin

Regs.ah := \$44; { Function number for IOCTL: Get Mode }

```

Regs.bx := Handle;
MsDos( Regs );                { Call DOS-Interrupt 21H }
GetMode := Regs.dx            { Pass device attribute }
end;

```

```

{ ***** }
{ * SETRAW : Change a character driver into RAW-Mode * }
{ * Input : the handle passed must be connected with * }
{ *         addressed device * }
{ * Output : none * }
{ ***** }

```

```

procedure SetRaw(Handle : integer);

```

```

var Regs : Registers;          { register-Variable for Interrupt call }

```

```

begin

```

```

  Regs.ax := $4401;            { Function number for IOCTL: Set Mode }
  Regs.bx := Handle;
  Regs.dx := GetMode(Handle) and 255 or 32;    { new device attribute }
  MsDos( Regs );                { Call DOS-Interrupt 21H }
end;

```

```

{ ***** }
{ * SETCOOKED : Change a character driver into the COOKED-Mode * }
{ * Input      : the handle passed must be connected with the * }
{ *              device addressed * }
{ * Output      : none * }
{ ***** }

```

```

procedure SetCooked(Handle : integer);

```

```

var Regs : Registers;          { register-Variable for Interrupt call }

```

```

begin
  Regs.ax := $4401;           { Function number for IOCTL: Set Mode }
  Regs.bx := Handle;
  Regs.dx := GetMode(Handle) and 223;      { new device attribute }
  MsDos( Regs );               { Call DOS-Interrupt 21H }
end;

```

```

{*****}
{ * TESTOUTPUT : Output a Test-String 1000 times on the Standard      * }
{ *               output device                                         * }
{ * Input       : none                                                 * }
{ * Output      : none                                                 * }
{*****}

```

```

procedure TestOutput;

```

```

var Regs : Registers;           { register-Variable for Interrupt call }
    LoopCnt : integer;         { Loop variable }
    Test : string[9];          { The Test-String for output }

```

```

begin
  Test := 'Test.... ';
  Regs.bx := STANDARDOUT;      { output on the Standard output device }
  Regs.cx := 9;                { Number of characters }
  Regs.ds := Seg(Test);        { Segment address of the text }
  Regs.dx := OfS(Test)+1;      { Offset address of the text }
  for LoopCnt := 1 to 1000 do
    begin
      Regs.ah := $40;           { Write function number for handle }
      MsDos( Regs );           { Call DOS-Interrupt 21H }
    end;
  end;

```

```
writeln;  
end;
```

```
{ *****  
*                                     *  
*                               MAIN PROGRAM                               *  
*                                     *  
* ***** }
```

```
begin  
  ClrScr;                                { Clear screen }  
  writeln('RAWCOOK (c) 1987 by Michael Tischer'#13#10);  
  writeln('The Console driver is now in RAW-Mode. Control keys such as  
<Ctrl><C>');  
  writeln('are not recognized during output. Press a key to display a text on  
'#13#10);  
  writeln('the screen, and try stopping the display by pressing <Ctrl><C>');  
  Keys := ReadKey;                        { wait for key }  
  SetRaw(STANDARDIN);                     { Console driver in RAW mode }  
  TestOutput;                             { Output Test-String 1000 times }  
  ClrScr;                                 { Clear Screen }  
  while KeyPressed do  
    Keys := ReadKey;                      { Empty keyboard buffer }  
    writeln('The Console driver is now in COOKED mode. Control keys such as');  
    writeln('<CTRL><C> are recognized during output');  
    writeln('Press a key to start, then press <Ctrl><C> to stop the display');  
    Keys := ReadKey;                      { Wait for key }  
    SetCooked(STANDARDIN);  
    TestOutput;                           { Output Test-String 1000 times }  
  end.
```