

Occam-Programmbeispiel

(nach Heinz Ebert: Occamierung auf Transputern, c't 1988, 1-4,6)

```
PROC INMOS_ENTRY_POINT()
  VAL 0.0    IS 0.0    (REAL32): -- einige benannte Konstanten, damit die
  VAL 0.5    IS 0.5    (REAL32): -- Typangabe (REAL32) nicht immer wieder
  VAL 0.0625 IS 0.0625 (REAL32): -- im Programm stehen muss ...
  VAL M0.75  IS -0.75  (REAL32):
  VAL 01.0   IS 1.0    (REAL32):
  VAL 04.0   IS 4.0    (REAL32):
  VAL 05.0   IS 5.0    (REAL32):
  VAL 016.0  IS 16.0   (REAL32):
  VAL MAXX   IS 320:   -- Breite
  VAL MAXY   IS 200:   -- Hoehe
  VAL MAXITER IS 40:   -- max. Anz. Iterationen
  VAL RLR    IS -2.1   (REAL32): -- Realteil linker Rand
  VAL RRR    IS 01.0:  -- Realteil rechter Rand
  VAL IOR    IS 1.15  (REAL32): -- Imaginaerteil oberer Rand
  -- eine Iteration
PROC ITERATE(VAL REAL32 xc,yc, INT k)
  REAL32 x,x2,y,y2:
  SEQ
    x :=0.0
    x2:=0.0
    y :=0.0
    y2:=0.0
    k :=0
  WHILE (((x2+y2)<=04.0)AND(k<MAXITER))
    SEQ
      y :=x*y
      y :=(y+y)+yc
      x :=(x2-y2)+xc
      x2:=x*x
      y2:=y*y
      k :=k+1
  :
  -- Ende des Prozesses ITERATE
  --
  -- Eine ganze Zeile erzeugen
PROC PRODUCE(VAL REAL32 fl,yc, [ ]BYTE vektor)
  REAL32 xc:
  SEQ
    xc:=RLR -- beginne am linken Rand
  SEQ ix=0 FOR MAXX
    INT k: -- Vereinbarungen fuer Teilprozess
    REAL32 x1,y2:
    SEQ
      y2:=yc*yc
      x1:=xc+01.0
    IF
      (xc > M0.75)
      REAL32 r,s:
      SEQ
        r:=(xc*xc)+y2
        SQRTP(s,(r-(0.5*xc))+0.0625)
      IF -- Test auf Zykloide
        (016.0*(r*s))>(((05.0*s)-(04.0*xc))+01.0))
        ITERATE(xc,yc,k)
      TRUE -- im Inneren (Rechnung ueberfluessig)
        k:=MAXITER
    TRUE
    IF -- Test auf Kreis
      (((x1*x1)+y2)>0.0625)
      ITERATE(xc,yc,k)
    TRUE
      k:=MAXITER
    vektor[ix]:=BYTE k
    xc:=xc+fl
  :
  -- Ende des Prozesses PRODUCE
```

```

-- "Hauptprozess"
PROC DOIT()
  VAL HALBE IS (MAXY-1)/2:
  VAL HALB1 IS HALBE+1:
  CHAN OF ANY aus, ein:
  PLACE aus AT #80000000:  -- Link 0 des Transputers
  [MAXX] BYTE v1,v2:
  REAL32 flx,fly,yc:
  SEQ
    yc:=IOR
    flx:=(RRR-RLR)/(REAL32 TRUNC(MAXX-1))
    fly:=(IOR+IOR)/(REAL32 TRUNC(MAXY-1))
    PRODUCE(flx,yc,v1)      -- 1. Zeile in 1. Puffer
    SEQ iy=0 FOR HALB1
      SEQ
        yc:=yc-fly
        PAR
          aus ! [v1 FROM 0 FOR MAXX]
          PRODUCE(flx,yc,v2)
        IF
          (iy<HALBE)      -- noch nicht letzte Zeile
            SEQ
              yc:=yc-fly
              PAR
                aus ! [v2 FROM 0 FOR MAXX]
                PRODUCE(flx,yc,v1)
            TRUE
            SKIP
          aus ! [v2 from 0 FOR MAXX]
    :
  DOIT()  -- der eigentliche Start
  :

```

Dazu gehört ein Programm, das auf dem Host-PC von Link 0 die Daten entgegennimmt und grafisch darstellt.