

# PRESEK

List za mlade matematike, fizike, astronome in računalnikarje

ISSN 0351-6652

Letnik 13 (1985/1986)

Številka 3

Strani 185-192

Bojan Mohar:

## OXFORD PASCAL

Ključne besede: računalništvo, programski jeziki, pascal.

Elektronska verzija: <http://www.presek.si/13/785-Mohar.pdf>

© 1986 Društvo matematikov, fizikov in astronomov Slovenije

© 2010 DMFA - založništvo

Vse pravice pridržane. Razmnoževanje ali reproduciranje celote ali posameznih delov brez poprejšnjega dovoljenja založnika ni dovoljeno.

# RAČUNALNIŠTVO

## OXFORD PASCAL

Trenutno najboljši prevajalnik za pascal na računalniku Commodore 64 je Oxford pascal. Dobiti se da tako verzijo na disketi kot na kaseti. Njegova velika prednost je v tem, da se skoraj brez izjem drži standardov. Razen tega vsebuje ukaze za delo z grafiko in zvokom. Njegov operacijski sistem je enostaven, a dovolj močan, da je delo z njim kar se da udobno.

Oxford pascal lahko uporabljamo na dva načina. Pri prvem, enostavnejšem, sta v pomnilniku istočasno tako prevajalnik kot naš program. Ta način je idealen za pisanje krajših programov, na primer takih, kot jih najdemo v učbenikih. Poleg velikosti programa je edina omejitev, ki jo postavlja ta način, da ne moremo uporabljati nekaterih ukazov za delo z disketo enoto. Pri pisanju daljših, profesionalnih programov pa bomo izbrali način, pri katerem je prevajalnik ves čas na disketi. V nadaljevanju bomo v glavnem opisovali le prvi način. (Kaj več o diskretnem načinu pa bomo povedali kdaj drugič.)

### 1. Uporaba prevajalnika

Najprej naložimo program za nalaganje prevajalnika:

```
LOAD "*" , 8
```

Ko dobimo sporočilo READY, ukažemo

```
RUN
```

in zopet počakamo na sporočilo READY. Tokrat čakamo malo dlje. Sedaj se namesto v basicovem nahajamo v pascalovem operacijskem sistemu. Oglejmo si, kako ga uporabljamo.

Program v pascalu, ki ga želimo vtipkati, ima tako kot programi v basicu oštevilčene vrstice. Torej začnemo pisati:

```
100 program test;
```

Ko pritisnemo RETURN, je vrstica vnesena, sistem pa nam sam napiše številko nove vrstice (110), da nam olajša delo. Če želimo končati z vnašanjem programa, pritisnemo le RETURN in sistem nam ne bo več vsiljeval novih številok vrstic.

Program popravljamo in spreminjamo na enak način kot programe v basicu.

Pascalov operacijski sistem uboga mnoge ukaze. Najpomembnejše bomo na kratko opisali.

- AUTO izključi avtomatični prikaz številke vrstic pri vnašanju programa.
- AUTO n spremeni razmik med številkami vrstic pri avtomatičnem vnašanju na n. Začetna vrednost razmika je  $n = 10$ .
- LIST obm izpiše del ali celoten program na ekran. Dodatek 'obm' določa območje, ki naj se izpiše. Ima enako obliko kot območje v basicu. Primer:

LIST	izpis celega programa
LIST 210	izpis vrste 210
LIST 100—	izpis vseh vrst od 100 dalje
LIST —100	izpis vseh vrst do vključno 100
LIST 100—200	izpis vrst s številkami med 100 in 200

Opomba: Tudi območja pri drugih ukazih, ki jih bomo navedli v nadaljevanju, imajo enako obliko in pomen.

- UPPER vključi velike črke.
- LOWER vključi male črke.
- NEW zbriše naš program v pomnilniku.
- DISK vključi disketni način (prevajalnik na disku).
- NUMBER n,m,k ponovno oštevilči vrstice. Začne z vrstico n, ki bo imela po novem številko m, korak med zaporednimi vrsticami pa bo enak k.
- FIND\*niz\*,obm v danem območju 'obm' (glej ukaz LIST) poišče (in izpiše) vse vrstice, v katerih se nahaja zaporedje znakov 'niz'. Primer;

FIND\*VAR\*, 100—500

- CHANGE\*niz1\*niz2\*, obm v danem območju nadomesti nize 'niz1' z nizom 'niz2'.

- DELETE obm izbriše vrstice, določene z 'obm'.
- PUT ime shrani program na disketo. Primer:

PUT TEST

je analogno basicovemu SAVE "TEST", 8.

GET ime	prebere program z diskete.
RUN (ali R)	izvede naš program. Če svojega programa še nismo prevedli, se najprej opravi prevajanje.
L	prevede program in ga hkrati izpisuje na ekran.
P	prevede program in ga hkrati izpisuje na tiskalnik.
DUMP obm	izpiše program na tiskalnik.

Razen tega lahko uporabljamo še nekatere basicove ukaze, na primer PRINT, CMD, POKE, SYS. Za direktorij na disketi uporabimo LOAD "\$", 8 in LIST. Pri tem seveda izgubimo svoj program v pomnilniku.

## 2. Nekateri dodatni podprogrami

Oglejmo si le tri podprograme. Prvi je VDU. Njegova oblika je naslednja:

VDU (vrstica, kolona, znak)

povzroči pa, da se v ustrezni vrstici in koloni izpiše dani znak. Gornji levi kot pomeni vrstico 0 in stolpec 0.

Funkcija GETKEY prebere znak s tipkovnice. Če ni bila pritisnjena še nobena tipka, vrne vrednost CHR(0). Tako na primer zanka

```
repeat ch := getkey until ch <> chr (0)
```

čaka, dokler ne pritisnemo katerekoli tipke.

Funkcija RANDOM vrne naključno število med 0 in 255.

## 3. Zvok in grafika

Prevajalnik ne bi bil vreden počenega groša, če nam ne bi omogočal dela z grafiko visoke ločljivosti in z zvokom. Za zvok imamo na voljo tri podprograme. Prvega pokličemo takole:

ENVEL (glas, naraščanje, upadanje, nivo zvoka, izzven)

Z njim določimo obliko zvoka. Vsi parametri so celoštevilski: glas ima vrednosti 1, 2 ali 3 in pove, za kateri glas gre. Ostali parametri pa imajo vrednosti med 1 in 15 in opisujejo vrsto zvoka. Drugi program je

VOICE (glas, frekvenca, oblika zvočnih valov, trajanje)

Glas je spet med 1 in 3, ravno tako oblika valov (trikotna, žagasta, pravokotna in šum). Frekvenca je med 0 in 65535 in jo dobimo tako, da frekvenco, izraže-

no v Hz, delimo s številom 0.059604645. Tabela frekvenc, ki ustrezajo posameznim glasbenim notam, najdemo v priločniku, ki ga dobimo skupaj z računalnikom. Trajanje je tudi število med 1 in 65535. Trajanje zvoka v glasbi je odvisno od tempa. Normalna vrednost za noto "četrtinko" bo okrog 500, za polovinko dvakrat več itd. Tretji podprogram določi glasnost (število med 1 in 15) in ga pokličemo z

#### VOLUME (glasnost)

S podprogramom ENVEL določimo obliko zvoka, z VOLUME glasnost, z VOICE pa zvok aktiviramo. Oblika zvoka ostane nespremenjena do naslednje uporabe podprograma ENVEL. Morda ni odveč omeniti, da ima vsak od treh glasov lahko svojo obliko.

Iz dosedanjega opisa zvoka bo le malokdo znal sprogramirati igranje na klavir. Priporočamo kakšno knjigo, v kateri je opisano generiranje zvoka na C-64, lahko tudi v basicu. Tu si oglejmo le nekaj parametrov za posamezne instrumente:

klavir:	envel (1, 0, 9, 0, 0),	oblika valov pravokotna
flavta:	envel (1, 6, 0, 0, 0),	oblika valov trikotna
orgle:	envel (1, 0, 0, 15, 0),	oblika valov trikotna
ksilofon:	envel (1, 0, 9, 0, 9),	oblika valov trikotna
violina:	envel (1, 5, 8, 5, 9),	oblika valov žagasta

#### Zaporedje stavkov:

envel (1, 0, 9, 0, 0); volume (15);  
voice (1, 2500, 1, 5000);

torej izvabi iz računalnika zvok, podoben pritisku na tipko malo slabšega klavirja.

Oglejmo si še podprograme za grafiko. Že v tekstovnem načinu imamo tri ukaze:

BORDER (barva)	določi barvo okvirja; med 0 in 15
SCREEN (barva)	določi barvo zaslona; med 0 in 15
PEN (barva)	določi barvo znakov; med 0 in 15

Posamezne vrednosti barv predstavljajo naslednje odtenke:

0 .... črna	3 .... turkizna
1 .... bela	4 .... vijoličasta
2 .... rdeča	5 .... zelena



6 .... modra	11 .... siva 1
7 .... rumena	12 .... siva 2
8 .... oranžna	13 .... svetlo zelena
9 .... rjava	14 .... svetlo modra
10 .... svetlo rdeča	15 .... siva 3

Za risanje uporabljamo grafiko visoke ločljivosti. Najprej z ukazom

PLOT (0, 0, 0, 0, 0)

zahtevamo, naj se napravi čista slika visoke ločljivosti, ki pa je zaenkrat še ne vidimo. S klicem

HIRES (1)

pa se ta slika zares pojavi na zaslonu. Če smo pred tem že kaj risali v grafiki visoke ločljivosti, bodo narisani liki še vedno vidni. Z ukazom

PLOT (1, 0, 0, 0, 0)

pa zaslon postane čist. Lahko mu še določimo barvo:

PAPER (barva)

kjer je barva med 1 in 15. Barvo peresa (črt) pa izberemo z

INK (barva)

Če se želimo vrniti iz grafike visoke ločljivosti v navadni, tekstovni način, zopet uporabimo podprogram HIRES, tokrat z argumentom 0:

HIRES (0)

Še en ukaz je, s katerim določamo grafični način. Klik

WINDOW (vrstica)

razdeli ekran na dva dela: od vrstice 0 do 'vrstica' je visoka ločljivost, v spodnjem delu ekrana pa imamo tekstovni način.

V grafičnem načinu z visoko ločljivostjo ima zaslon našega računalnika 320 x 200 točk. Vsaka od teh točk je lahko osvetljena ali pa ne in s tem lahko tvorimo najrazličnejše slike. Vsaka točka ima dve koordinati (x, y),  $0 \leq x < 320$ ,  $0 \leq y < 200$ . Točka (0, 0) je v spodnjem levem kotu, smer osi x je od

leve proti desni, smer osi y pa od spodaj navzgor.

Za risanje črt uporabljamo podprogram PLOT. Klic

PLOT (p, x1, y1, x2, y2)

dela črto od (x1, y1) do točke (x2, y2). Če je vrednost prvega parametra enaka  $p = 2$ , se črta nariše, če pa je  $p = 3$ , se črta zbriše. Klic PLOT (2, x, y, x, y) osvetli točko (x, y). Funkcijski podprogram

EXAMINE (x, y)

vrne vrednost 1, če je točka (x, y) osvetljena, sicer pa vrne 0.

Za konec sestavimo še program, ki nariše lik, sestavljen iz koncentričnih kvadratov s središčem v točki (100, 100).

```
program koncentricni;
  (* risanje koncentričnih kvadratov *)
const   sx = 100;
        sy = 100;
var     i : integer;

procedure kvadrat (a : integer);
  (* nariše kvadrat s središčem (sx, sy) in stranico a *)
var     x, y, a2 : integer;
begin
  a2 := a div 2;
  x := sx - a2;
  y := sy - a2;
  plot (2, x, y, x + a, y);
  plot (2, x + a, y, x + a, y + a);
  plot (2, x + a, y + a, x, y + a);
  plot (2, x, y + a, x, y);
end; (* kvadrat *)

procedure zacetek;
begin
  paper (5); ink (1);
  plot (0, 0, 0, 0, 0);
  plot (1, 0, 0, 0, 0);
  hires (1)
end;

procedure konec;
```





```

program izpis;
  (* izpis na tiskalnik *)
var ch : char;
  ime : packed array [1 .. 15] of char;
  disk, tiskalnik : text;
begin
  write ('Vnesi ime datoteke : ');
  readln (ime);
  reset (disk, ime);
  rewrite (tiskalnik, 4, 0);
  while not eof (disk) do begin
    read (disk, ch);
    write (tiskalnik, ch);
    if eoln (disk) then begin
      readln (disk);
      writeln (tiskalnik)
    end
  end;
  writeln;
  writeln ('Izpis končan')
end.

```

*Bojan Mohar*