



(Ponoviti LV 11 iz Programiranja I – pokazivači i reference)

1. Deklaracija pokazivača

Pokazivač je varijabla koja čuva memorijsku adresu objekta na kojeg pokazuje. Iako na prvi pogled svrsishodnost pokazivača nije očita, u C/C++ jeziku njihova primjena je vrlo rasprostranjena, jer pružaju praktički neograničenu fleksibilnost u pisanju programa.

Pokazivač se može deklarirati tako da pokazuje na bilo koji tip podatka. U deklaraciji se navodi tip podatka na koji pokazivač pokazuje, a ispred imena se stavlja unarni operator zvjezdica (*). Naredba:

```
int *pok;
```

deklarira pokazivač **pok** na objekt tipa **int**. Smisao pokazivača ilustrirat ćemo sljedećim kodom:

```
int *pok;           // pokazivač na varijablu tipa int
```

```
int n = 5;
```

```
pok = &n;          // usmjeri pokazivač na n
```

Prvom naredbom se deklarira varijabla **pok** kao pokazivač na broj tipa **int**. Zatim se deklarira varijabla **n** te joj se pridružuje vrijednost 5. Na kraju se varijabli **pok**, pomoću adresnog operatora **&** pridružuje vrijednost memorijske adrese na kojoj je pohranjena varijabla **n**. Sada vrijednost varijable **n** možemo dohvaćati izravno: `printf("%d", n);` ili preko pokazivača: `printf("%d", *pok);`

Zadatak za vježbu:

1. Napišite C program s deklaracijom neke cjelobrojne varijable i deklaracijom nekog pokazivača na cjelobrojni objekt. Cjelobrojnoj varijabli dodijelite neku vrijednost. Preusmjerite pokazivač na cjelobrojni varijablu. Ispišite slijedeće podatke na ekran:

- a) vrijednost varijable izravno,
- b) vrijednost varijable pomoću pokazivača,
- c) memorijsku adresu deklarirane varijable,
- d) memorijsku adresu pokazivača.

Sve operacije koje su dozvoljene izravno na cjelobrojnoj varijabli, mogu se provesti i preko pokazivača. Tako naredba:

```
*pok += 5;
```

povećava vrijednost na memorijskoj lokaciji na koju pokazuje pokazivač za iznos 5. Ako se izostavi znak * (zvjezdica), npr.:

```
pok += 5;
```

to znači da se pokazivač preusmjerava na memorijsku adresu koja je za 5 blokova veća od prije postavljene, pri čemu je veličina bloka definirana prostorom koji zauzima **int** varijabla.

Budući da brojčane konstante nemaju svoj prostor u memoriji, slijedeća naredba nema smisla:

```
float *pi = &3.14;
```

Međusobne operacije s pokazivačima na različite tipove podataka nisu preporučljive i redovito su nedozvoljene.

Pokazivač na tip `void` može se preusmjeriti na objekt bilo kojeg tipa:

```
int n = 5;
int *pokn = &n;
float x = 10.27;
float *pokx = &x;
void *pokvoid;

pokvoid = &n;
pokvoid = pokx;
```

Međutim, pokazivaču na `void` ne možemo izravno dodijeliti vrijednost:

```
*pokvoid = x;
```

Pokazivači bez obzira na koji tip podatka pokazuju, na istom računalu uvijek zauzimaju jednake memorijske prostore.

Zadatak za vježbu:

2. Napišite C program koji će na ekran ispisati veličinu pokazivača na slijedeće tipove podataka: `char`, `int`, `float`, `long`, `double`. Za provjeru veličine memorije koju neka varijabla zauzima u memoriji treba koristiti operator `sizeof`, npr. `sizeof(int*)`.

2. Veza između pokazivača i polja

U programskom jeziku C pokazivači i polja su međusobno čvrsto povezani. Iako to kod dohvaćanja preko indeksa nije očito, članovi polja dohvaćaju se u biti preko pokazivača. Naredbom:

```
float x[5];
```

deklarira se jednodimenzionalno polje objekata koje se sastoji od pet članova tipa `float`. Pri tome samo ime `x` ima smisao pokazivača na prvi član polja `x[0]`. Prilikom dohvaćanja članova polja, prevoditelj će vrijednost indeksa pribrojiti pokazivaču na prvi član; tako će naredba

```
float a = x[2];
```

biti zapravo prevedena kao

```
float a = *(x + 2);
```

Ova zadnja naredba može se interpretirati na slijedeći način: uzmi adresu prvog člana polja, povećaj ju za dva, pogledaj što se nalazi na toj adresi te pridruži vrijednost na toj adresi varijabli `a`.

Pristupi članovima polja preko indeksa ili preko pokazivača potpuno su ekvivalentni.

Zadatak za vježbu:

3. Napišite C program koji će s tipkovnice učitati `n` cijelih brojeva u polje `V[10]`. Pristup članovima niza, pri učitavanju, treba napraviti preko pokazivača. Izračunajte i ispišite srednju vrijednost članova niza, ali tako da tim članovima pristupate preko pokazivača. Napravite ispis sljedećih podataka za sve članove polja tako da dobijete tablicu: indeks člana, vrijednost člana niza, memorijska adresa člana niza.

Korištenje 2D-polja pomoću pokazivača:

```
int i, j;
float A[10][10], *p;
p=&A[0][0];
printf ("Unesite elemente niza:");
for (i=0;i<m;i++)
    for (j=0;j<n;j++)
        scanf ("%f", (p+i*n+j));
```

3. Pokazivači i funkcije

Pokazivači se često upotrebljavaju za prijenos podataka u funkciju preko adrese, a to se postiže najavom argumenata funkcije kao pokazivača. Na taj način funkcija može mijenjati vrijednost prenesenih varijabli:

```
void zamjena (int *c, int *d)
{
    int t;
    t=*c;
    *c=*d;
    *d=t;
}
```

Zadatak za vježbu:

4. Napisati program koji s tipkovnice učitava cijele brojeve: $2 < m < 11$ i $2 < n < 11$, zatim učitava $m \times n$ realnih brojeva u 2D polje A pomoću funkcije ucitaj2D. Ispisati 2D polje pomoću funkcije ispis2D (koristiti pokazivače). Napraviti funkciju dijeli2D povratnog tipa void koja prima 3 parametra: pokazivač na A (prima adresu nultog elementa polja A), broj redaka m i broj stupaca n. Funkcija svaki element polja A dijeli sa 2. Ispisati ponovno polje A pomoću ispis2D.

4. Pokazivači i stringovi

Ispis i brojanje znakova stringa pomoću pokazivača:

```
char string[]="Ovo je probni ispis!";
char *s;
int br=0;
s=string;
while (putchar(*s++)) br++;
printf ("\nU stringu je %d znakova.\n",br);
```