

Programski jezici i strukture podataka

4

POKAZIVAČI

Pokazivači

- Referenca na podatak tipa objekta ili na funkciju.
- Primene:
 - pri definisanju funkcija koje se pozivaju po referenci
 - pravljenje dinamičkih struktura
 - spregnute liste
 - stabla...

Pokazivači

- Pokazivač je adresa u memoriji, broj.
 - ukazuje na lokaciju u memoriji i tip objekta ili funkcije
 - sadržaj pokazivačke promenljive je adresa (lokacija u memoriji).
- Pokazivačke promenljive pokazuju na druge promenljive ili na početak memorijskog bloka
 - pokazivači na promenljive tipa int, float, itd.

Adrese u memoriji

- Operativna memorija je niz memorijskih lokacija koje su numerisane celim brojevima $0, 1, 2, m-1$, pri čemu je m kapacitet memorije.
- Brojevi pridruženi memorijskim lokacijama nazivaju se ***adresama***.
- Najmanja memorijska lokacija koja može samostalno da se adresira je obično jedan bajt koji se sastoji od 8 bitova.
- Podaci različitih tipova smeštaju se u jedan ili više uzastopnih bajtova.
- Na primer char u 1 bajt, short int u 2 bajta, long int i float u 4 bajta, a double u 8 bajtova.

Pokazivač

- ***Pokazivač*** je prost podatak u koji može da se smesti adresa neke lokacije u memoriji. Pokazivači obično zauzimaju 4 bajta.
- Broj bajtova koje zauzima pokazivač zavisi od mogućeg opsega adresa na datom računaru, a ne od broja bajtova koliko zauzimaju pokazivani podaci.
- Kada neki podatak zauzima više bajtova, pod adresom podatka podrazumeva se bajt sa najmanjim rednim brojem.

Deklarisanje pokazivača

- Sintaksa deklarisanja pokazivača na objekat koji nije niz:

```
type * [lista-kvalifikatora-  
tipa] ime [ = inicijalizator ] ;
```

- U deklaracijama * znači pokazivač na.
- Identifikator *ime* deklarise se kao objekat tipa *tip **, ili kao pokazivač na tip *tip*.

Definisanje pokazivača

- Opis tipa na početku deklarativne naredbe označava tip **pokazivanog** podatka.
- Pokazivači se definišu, kao i svi ostali podaci, uobičajenim naredbama za definisanje , s tim što se u nazivu podatka dodaje modifikator * ispred identifikatora podatka:

*** identifikator_pokazivaca**

- identifikator standardnih prostih tipova,
- identifikator tipa ranije uveden naredbom typedef
- opis tipa koji definiše programer (na primer opis nabiranja – enum)

Referenciranje

- Unarni operator `&` daje adresu promenljive
- Izraz `p=&a` dodeljuje adresu promenljive `a` promenljivoj `p`, pa sada `p` pokazuje na `a`
- Da bi se odštampana vrednost pokazivača koristi se konverzija `%p`.

Referenciranje

- Adresa nekog podatka u memoriji može da se dobije pomoću prefiksnog unarnog operatora &.
- Operand operatora & mora da bude podatak koji se nalazi u memoriji.
- Operator & **ne sme** da se primeni na prolazne podatke za koje je prilikom definisanja traženo da se smeste u procesorske registre (modifikator tipa register)

Dereferenciranje

- Primenom unarnog operatora * može se posredno pristupiti nekom podatku pomoću memorijske adrese.
- Dohvatanje podatka posredno, pomoću adrese, naziva se i ***indirektnim adresiranjem***.
- Zato se i sam operator * ponekad naziva operatorom indirektnog adresiranja.

Specijalna konstanta NULL

- Konstanta koja se nalazi u `stdio.h`
- Ako pokazivačka promenljiva ima vrednost NULL, onda ne pokazuje ni na šta.
- Primer:

```
int *p;  
p = NULL;
```
- Neinicijalizovana vrednost NEMA NULL!
 - mora se eksplicitno inicijalizovati na NULL

Pokazivači na vrednost NULL

- Pokazivač na vrednost `NULL` (null pointer) dobija se kada se konvertuje ***konstantan pokazivač*** na vrednost `NULL` u tip pokazivač (***z19.c***).
- ***Konstanta pokazivača*** na vrednost `NULL` jeste celobrojan konstantni izraz vrednosti 0, konvertovan na tip `void *`.

Pokazivač na tip void

- Pokazivač na tip void predstavlja pokazivač tipa void *.
- Objekti tipa void ne postoje, zato se pokazivač ovog tipa koristi kao pokazivač opšte namene.
- On može da predstavlja adresu bilo kog objekta, ali ne i njegov tip.
- Da bi pristupili objektu u memoriji ovakav pokazivač se uvek kastuje na odgovarajući tip.

Notacija

- Notacija `int *p` može da se tumači ovako: ako je `*p` podatak tipa `int`, `p` mora da je pokazivač na takve podatke.
- Treba posebno naglasiti da u slučaju definisanja više pokazivača jednom naredbom, modifikator `*` stavlja se ispred svakog identifikatora.

Inicijalizacija pokazivača

- Pokazivačke promenljive nakon deklaracije imaju nedefinisanu početnu vrednost.
- Pokazivač se može inicijalizovati pomoću sledećih vrsta inicijalizacije:
 - konstantan pokazivač na vrednost NULL
 - Pokazivač na isti tip
 - Pokazivač na tip void
- Inicijalizator pri definisanju pokazivača treba da bude adresa u operativnoj memoriji. Na primer, naredbom:

```
double x, *px=&x;
```
- definiše se podatak x tipa double i pokazivač px na podatke tipa double.
- Pokazivač px se inicijalizuje adresom (&x) prethodno definisane promenljive x.

Nepromenljivi i nepostojani podaci i pokazivači

- Do sad definisani pokazivači su ***promenljivi pokazivači na promenljive podatke***.
- To znači da vrednost takvih pokazivača može da se promeni, a takođe, pomoću takvih pokazivača može da se promeni vrednost pokazivanog podatka.
- Ako se prilikom definisanja pokazivača, na početku naredbe za definisanje podataka dodaje modifikator **const** dobiće se ***promenljivi pokazivač na nepromenljive podatke***.
- Nepromenljivost se odnosi na pokazivane podatke, a ne na pokazivače same.

```
const tip *p;
```

- `p` je pokazivač na nepromenljive (const) podatke tipa `tip`.
- Vrednost pokazivača na nepromenljive podatke može da se promeni, ali pomoću njega ne može da se promeni vrednost podatka na koji on pokazuje.

Nepromenljivi i nepostojani podaci i pokazivači

- Nepromenljivi pokazivači se dobijaju korišćenjem modifikatora ***const** ispred samog identifikatora definisanog pokazivača.
- Vrednosti takvih pokazivača ne mogu da se promene kasnije u programu, pa moraju da se navedu i inicijalizatori.

tip ***const p = &k;**

- Nepromenljivi pokazivač na promenljive podatke.
- p je nepromenljivi (const) pokazivač (*) na podatke tipa tip.

Nepromenljivi i nepostojani podaci i pokazivači

```
const tip *const p = &k;
```

- *nepromenljivi pokazivač na nepromenljive podatke.*
- p je nepromenljivi (const) pokazivač (*) na nepromenljive (const) podatke tipa *tip*.