

Programiranje i programski jezici



Operacija adresiranja: &

- Unarna operacija & daje adresu promenljive.
- Vrednost izraza &x je adresa promenljive x, a ne njena tekuća vrednost.

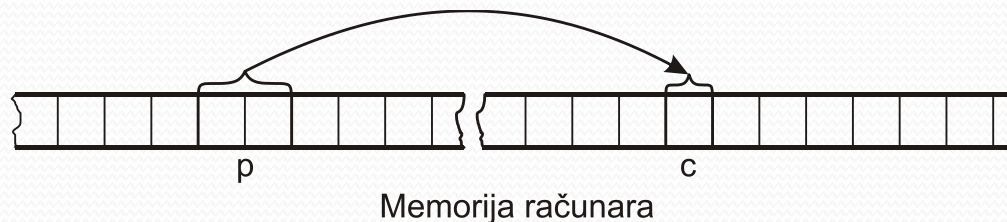
```
#include <stdio.h>
main()
{
    int x=7,y=10;
    printf("U main(): x=%d i &x=%p\n",x,&x);
    printf("U main(): y=%d i &y=%p\n",y,&y);
}
```

- izlaz: U main(): x=7 i &x=5693:0FDA
U main(): y=10 i &y=5693:0FDC

Operacija adresiranja: &

- Konverzaciona specifikacija **%p** u **printf()** obezbeđuje ispis adrese u obliku
- broj_segmenata:prirastaj

Pokazivači i adrese



- Pokazivačka promenljiva je promenljiva koja “pokazuje” na drugu promenljivu, odnosno sadrži adresu memorijske lokacije u kojoj se čuva ta promenljiva.

- Simbol * ispred imena promenljiva ukazuje da je ona pokazivačka promenljiva

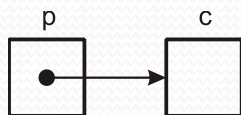
```
int *pi;
```

```
char *pc;
```

```
float *pf;
```

- Dobijanje adrese nekog objekta (operator **&**)

```
p = &c;           /* p pokazuje na c */
```



Pokazivači i adrese

- Operator & se primenjuje samo na objekte u memoriji
- Pristupanje objektu na koji pokazivač pokazuje (operator *, operator dereferenciranja)

```
int x = 1, y = 2;  
int *ip; /* ip je pokazivac na int */  
  
ip = &x; /* ip sada pokazuje na x */  
y = *ip; /* y je sada 1 */  
*ip = 0; /* x je sada 0 */
```

Primer 1

```
#include <stdio.h>
main() {
    int x = 3;
    int *px;
    printf("Adresa promenljive x je : %p\n", &x);
    printf("Vrednost promenljive x je : %d\n", x);
    px = &x;
    printf("Vrednost promenljive px je (tj. px) : %p\n", px);
    printf("Vrednost promenljive na koju ukazuje px je:%d\n", *px);
    *px = 6;
    printf("Vrednost promenljive na koju ukazuje px je: %d\n", *px);
    printf("Vrednost promenljive x je : %d\n", x);
}
```

Pokazivačke promenljive

```
int i,j, *pi1, *pi2;
```

Pokazivačke
promenljive

pi1



pi2



- Na početku promenljive pi1 i pi2 ne pokazuju ni na šta.

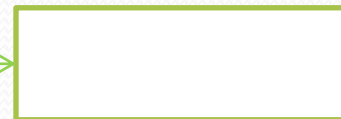
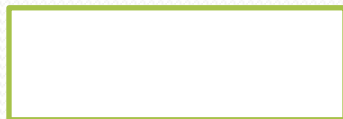
Operatori

Pokazivačke
promenljive

Promenljive

```
pi1=&i;
```

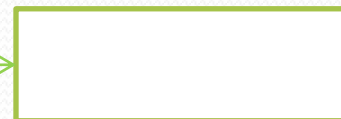
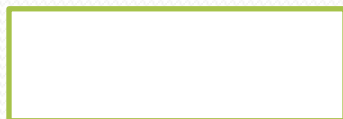
pi1



i

```
pi2=&j;
```

pi2



j

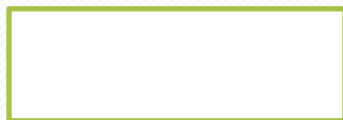
Pokazivačke promenljive

Operatori

`i=5;`

Pokazivačke
promenljive

pi1



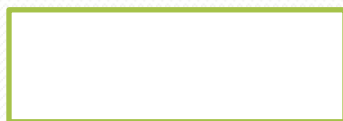
Promenljive

5

`i,*pi1`

`j=7;`

pi2



7

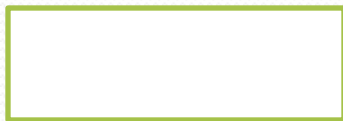
`j,*pi2`

Operatori

`i=j;`

Pokazivačke
promenljive

pi1

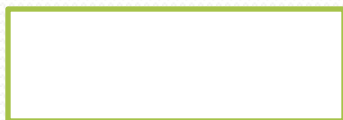


Promenljive

7

`i,*pi1`

pi2



7

`j,*pi2`

Pokazivačke promenljive

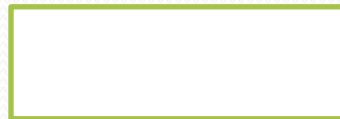
Operatori

Pokazivačke
promenljive

Promenljive

```
*pi1=(int)3.14+6
```

pi1



i,*pi1

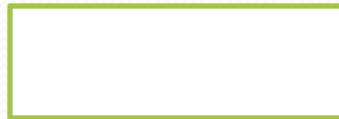
- Operiše se kao sa promenljivima bilo kog tipa

Operatori

Pokazivačke
promenljive

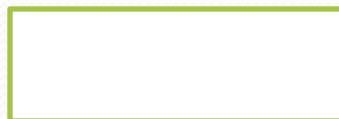
Promenljive

pi1



```
pi2=pi1;
```

pi2



i,*pi1,*pi2

Pokazivači i argumenti funkcija

- Pozvana funkcija ne može direktno da promeni promenljivu u funkciji koja je poziva
- Primer: Funkcija koja menja mesta dvema promenljivama

```
swap(x,y);  
  
...  
void swap(int a, int b) /* POGRESNO */  
{  
    int temp;  
  
    temp = a;  
    a = b;  
    b = temp;  
}
```

Pokazivači i argumenti funkcija

Promena se može postići slanjem pokazivača na promenljive koje treba promeniti

```
swap(&x, &y);
```

```
...
```

```
void swap(int *pa, int *pb) /* zamena *pa i *pb */
```

```
{  
    int temp;
```

```
    temp = *pa;
```

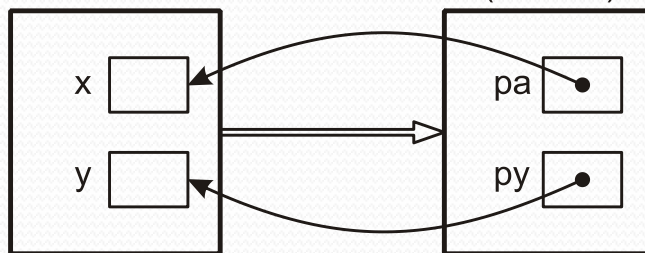
```
    *pa = *pb;
```

```
    *pb = temp;
```

```
}
```

Funkcija koja poziva

Pozvana funkcija
(zamena)



Shematski prikaz prosleđivanja pokazivača funkciji

Primer 2

```
#include <stdio.h>
void swap_wrong(int x, int y) {
    int tmp;
    printf("x : %p\n", &x); printf("y : %p\n", &y);
    tmp = x;
    x = y;
    y = tmp;
}
void swap(int* px, int* py) {
    int tmp;
    printf("px = %p\n", px); printf("py = %p\n", py);
    tmp = *px;
    *px = *py;
    *py = tmp;
}
```

Primer 2 (nastavak)

```
main()
{
    int x = 3, y = 5;
    printf("Adresa promenljive x je %p\n", &x);
    printf("Vrednost promenljive x je %d\n", x);
    printf("Adresa promenljive y je %p\n", &y);
    printf("Vrednost promenljive y je %d\n", y);
    swap_wrong(x, y);
    printf("Posle swap_wrong:\n");
    printf("Vrednost promenljive x je %d\n", x);
    printf("Vrednost promenljive y je %d\n", y);
    swap(&x, &y);
    printf("Posle swap:\n");
    printf("Vrednost promenljive x je %d\n", x);
    printf("Vrednost promenljive y je %d\n", y);
}
```

Zadatak 1

Napisati funkciju koja istovremeno vraća dve vrednosti -kolicnik i ostatak dva data broja.

Zadatak 1

```
#include <stdio.h>

void div_and_mod(int x, int y, int* div, int* mod) {
    printf("Kolicnik postavljam na adresu : %p\n", div);
    printf("Ostatak postavljam na adresu : %p\n", mod);
    *div = x / y;
    *mod = x % y;
}

main() {
    int div, mod;
    printf("Adresa promenljive div je %p\n", &div);
    printf("Adresa promenljive mod je %p\n", &mod);
    div_and_mod(5, 2, &div, &mod);
    printf("Vrednost promenljive div je %d\n", div);
    printf("Vrednost promenljive mod je %d\n", mod);
}
```