

Programiranje 1

Programski jezik C

2. čas

Mirko Spasić

Operatori

- U C-u postoji veliki broj operatora. Mogu biti **unarni** (imaju jedan argument) i **binarni** (dva argumenta).
- Unarni operatori mogu biti prefiksni i sufiksni.
- Binarni operatori su uvek infiksni
- Operatori imaju svoj prioritet i asocijativnost.

Izrazi

- Kombinovanjem promenljivih, konstanti i operatora dobijemo izraze
- Svaki izraz ima svoj tip i svoju vrednost.
- Tip izraza zavisi od tipova podizraza koji ga čine, kao i od operatora kojim se ti podizrazi povezuju.
- Ako su operandi neodgovarajućeg tipa, tada se implicitno konvertuju u odgovarajući tip, ako je to moguće. Koja će se konverzija izvršiti zavisi od operatora.

Primer konverzije

- Naredba dodele

```
int i = 5;
```

```
float f = 2.3;
```

```
f = i; /* f ce imati vrednost 5.0 */
```

- obrnuto:

```
int i = 5;
```

```
float f = 2.3;
```

```
i = f; /* i ce imati vrednost 2 */
```

Konverzije tipova

- Tip izraza se može eksplicitno promeniti tzv. **cast** operatorom (ispred izraza se u zagradi navede ime tipa u koji želimo da konvertujemo izraz).

(tip)<izraz>

float x;

x = 2.3+4.2; /* x ce imati vrednost 6.5 */

x = (int)2.3+(int)4.2; /*x ce imati vrednost 6 */

x = (int)2.3*4.5; /* x ce imati vrednost 9.0 jer zbog prioriteta operatora konverzije prvo ce biti izvršena konverzija broja 2.3 u 2 pa tek onda izvršeno množenje. */

x = (int)(2.3*4.5); /* x ce imati vrednost 10.0 */

Primer

- Celobrojno deljenje:

```
int x = 7, y = 2, z;
```

```
double w;
```

```
z = x / y; /* z ce imati vrednost 3, jer se vrši celobrojno  
deljenje */
```

```
w = x / y; /* w ce imati vrednost 3.0, jer se vrši  
celobrojno deljenje x sa y, pa se ceo broj 3  
konvertuje na 3.0 */
```

```
w = (1.0*x) / y; /* ovo je način da se izbegne celobrojno  
deljenje implicitnom konverzijom */
```

```
w = (0.0+x) / y; /* i ovo je način da se izbegne  
celobrojno deljenje implicitnom  
konverzijom */
```

```
w = (double)x / y; /* ovo je način da se izbegne  
celobrojno deljenje eksplicitnom  
konverzijom */
```

Aritmetički operatori

- Operatori **+**, **-**, *****, **/**, **%** nazivaju se aritmetički operatori.
- Ovi operatori se primenjuju na celobrojne i realne tipove.
- Operator **/** primenjen na cele brojeve daje celobrojni količnik.
- Operator **%** se primenjuje samo na cele brojeve i daje ostatak pri deljenju.

Aritmetički operatori

- Rezultat aritmetičkih operacija je izraz istog tipa kao i operandi.
- Ako operandi nisu istog tipa tada se vrši implicitna konverzija užeg u širi tip.
- Ako množimo dva broja tipa double, i rezultat smeštamo u proenljivu tipa int, tada će se tu izvršiti implicitna konverzija šireg u uži tip, pa će nam kompajler prijaviti upozorenje o mogućem gubitku informacija

Relacioni operatori

- Operatori `==`, `!=`, `<`, `>`, `<=`, `>=` nazivaju se relacioni operatori.
- Ovi operatori se primenjuju na celobrojne i realne tipove.
- Rezultat relacionih operacija je izraz celobrojnog tipa sa vrednošću 1 ako je relacija tačna, 0 u suprotnom.
- Ako operandi nisu istog tipa tada se vrši implicitna konverzija užeg u širi tip.

Logički operatori

- **!** – unarna negacija
- **&&** - logičko i
- **||** - logičko ili

- U C-u ne postoji logički tip.
- Svaki izraz koji se može porediti na jednakost sa nulom se u C-u smatra logičkim izrazom.
- Ako je vrednost izraza različita od 0, tada se on smatra logički tačnim. Izraz jednak nuli se smatra logički netačnim.
- Rezultat logičkih operacija je izraz celobrojnog tipa, jednak 0 ako je rezultat logičke operacije netačno, odnosno 1, ako je rezultat logičke operacije tačno.

Logički operatori - Primeri

- $5 \ \&\& \ 4$ - vrednost je 1, tj. tačno
- $0 \ || \ 10$ - vrednost je 1, tj. tačno
- $0 \ \&\& \ 5$ - vrednost je 0, tj. netačno
- $!1$ - vrednost je 0, tj. netačno
- $!9$ - vrednost je 0, tj. netačno
- $!0$ - vrednost je 1, tj. tačno
- $!(2>3)$ - vrednost je 1. tj. tačno

Operatori sa bočnim efektima

- Pojava da se prilikom izračunavanja nekog izraza menja vrednost neke promenljive naziva se bočni efekat (eng. side effect).
- Operatori koji imaju bočni efekat su operatori dodele i operatori uvećanja i umanjenja.
- Naredba dodele se u C-u predstavlja izrazom dodele, a njeno izvršavanje se zasniva na bočnim efektima operatora dodele.

Operatori dodele

- Operator proste dodele je `=`. Ne treba ga mešati sa relacionim operatorom `==`.
- Levi operand ovog operatora je leva vrednost (ime promenljive). Desni operand je proizvoljni izraz.
- Najpre se izračunava izraz na desnoj strani. Njegova vrednost se zatim po potrebi konvertuje u tip promenljive na levoj strani. Nakon toga se dobijena vrednost upisuje u memorijsku lokaciju koja je rezervisana za čuvanje te promenljive.

Operatori dodele

- Tip izraza dodele je tip promenljive na levoj strani. Vrednost izraza dodele je vrednost koja je dodeljena promenljivoj.
- Operatori složene dodele su $+=, -=, *=, /=, \% =$.
- Izraz oblika $a+=E$ je ekvivalentan izrazu $a=a+(E)$.
- Slično je sa ostalim operatorima.

Operatori umanjenja i uvećanja

- Unarni operatori **++** i **--** nazivaju se operatori uvećanja (inkrementacije) i umanjenja (dekrementacije) respektivno.
- Mogu biti prefiksni i sufiksni.
- U oba slučaja vrši se uvećanje promenljive za 1 ali izraz **++n** uvećava promenljivu n pre nego što se njena vrednost koristi, dok **n++** uvećava n nakon što se njena vrednost koristi. Tako se **x=++n;** razlikuje od **x=n++;** .
- Slično za operator **--**, s tim što je u pitanju umanjenje za jedan.

Prioritet i asocijativnost operatora

- Unarni operatori imaju viši prioritet od binarnih.
- Aritmetički operatori su višeg prioriteta od relacionih, a ovi višeg od logičkih.
- Operatori dodele su najnižeg prioriteta.
- Ako dva operatora imaju isti prioritet, onda se u obzir uzima asocijativnost, koja može biti s leva na desno, ili s desna na levo.
- Prioritet operatora se može promeniti korišćenjem zagrada ().
- Binarni operatori su leve asocijativnosti, osim operatora dodele koji imaju desnu asocijativnost. Unarni operatori su mahom desne asocijativnosti, ali ima izuzetaka.

Prosta naredba

- Prosta naredba se sastoji iz izraza za kojim sledi karakter ':'
- Naredba se izvršava tako što se izračunava izraz iz koga se sastoji uz sve bočne efekte koji se pri tom javljaju.
- Jasno je da prosta naredba ima smisla samo ako sadrži bar jedan bočni efekat.

Kontrola toka - IF

if (izraz)

 naredba1

else

 naredba2

- Naredba može biti prosta naredba a može biti i složena naredba (blok) koja se dobije kada se više prostih naredbi grupišu navođenjem vitičastih zagrada.

IF - Primer

- *Program ilustruje if i ispisuje ukoliko je uneti ceo broj negativan.*

```
#include <stdio.h>
int main() {
    int b;
    printf("Unesi ceo broj:");
    scanf("%d", &b);
    if (b < 0)
        printf("Broj je negativan\n"); //prosta naredba
    return 0;
}
```

IF - Primer

- Ulaz:
Unesi ceo broj:-5
Izlaz:
Broj je negativan
- Ulaz:
Unesi ceo broj:5
Izlaz:

Naredba IF - ELSE

- Else se odnosi na prvi neuparen if. Ako želimo drugačije moramo da navedemo vitičaste zagrade.

```
if (izraz) /* prvo if */
```

```
if (izraz1) naredba1; /* drugo if */
```

```
    else naredba2
```

- Ovo else se odnosi na drugo if, a ne na prvo if!

IF - ELSE

```
if (izraz) {  
    if (izraz1)  
        naredba1  
}  
else  
    naredba2
```

- Tek sada se else odnosi na prvo if!!!

ELSE IF konstrukcija

```
if (izraz1)
    iskaz1
else if (izraz2)
    iskaz2
else if (izraz3)
    iskaz3
else if (izraz4)
    iskaz4
else iskaz
```

Primer

- *Program ispituje znak broja.*

```
#include <stdio.h>
int main() {
    int b;
    printf("Unesi ceo broj : ");
    scanf("%d", &b);
    if (b < 0)
        printf("Broj je negativan\n");
    else if (b == 0)
        printf("Broj je nula\n");
    else
        printf("Broj je pozitivan\n");
    return 0;
}
```


Pogrešan primer dodela umesto poređenja

```
#include <stdio.h>
int main()
{
    int b;
    printf("Unesi ceo broj : ");
    scanf("%d", &b);
    /* Obratiti paznju na = umesto == Analizirati rad programa*/
    if (b = 0)
        printf("Broj je nula\n");
    else if (b < 0)
        printf("Broj je negativan\n");
    else
        printf("Broj je pozitivan\n");
    return 0;
}
```

- Šta se dešava ako unesemo broj -5?
- Biće ispisano "Broj je pozitivan". Zašto?