

Osnovi programiranja

priprema za prvi kolokvijum

Goran Rakić

februar 2005.

Napomena

Pred vama se nalaze beleške sastavljene neposredno pred izlazak na prvi kolokvijum iz Osnova programiranja. Beleške svakako nisu kompletne, niti je to i planirano da budu. Takođe, ne postoji nikakva garancija da su iznesene informacije uopšte tačne iako ja trenutno verujem da jesu. Izvori su dati na kraju. Sve komentare (izuzev onih uvredljive prirode) ću rado primiti.

1 Brojni sistemi

Binarni (2): 0, 1

Oktalni (8): 0, 1, 2, 3, 4, 5, 6, 7

Dekadni (10): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Heksadecimalni (16): 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Konverzija iz $2, 8, 16 \Rightarrow 10$ se vrši pronalaženjem sume umnožaka cifre i stepenovane osnove na broj trenutne cifre počev sa desne strane umanjeno za jedan za svaku cifru.

Konverzija iz $10 \Rightarrow 2, 8, 16$ se vrši celobrojnim deljenjem brojem 2, 8 ili 16. Ostaci pri deljenju daju cifre u traženom zapisu. Poslednji ostatak pri deljenju (kada količnik postane 0) je prva cifra u novom zapisu i tako redom do prvog ostatka koji je poslednja cifra.

Za konverziju iz $2 \Rightarrow 16$ najpre treba podeliti cifre u binarnom zapisu u grupe od po četiri cifre (dopuniti nulama sa leve strane po potrebi, dakle $23_{(10)} = 10111_{(2)}$ postaje 00010111), a onda svaku grupu prevesti u heksadecimalni zapis. Rezultat se dobija cifra po cifra. Pomenuti primer dakle postaje $0001_{(2)} = 1_{(16)}$ i $0111_{(2)} = 7_{(16)}$, odnosno $10111_{(2)} = 17_{(16)}$. Za konverziju u obrnutom smeru, obrnuti ceo proces.

I na kraju, konverzija iz $2 \Rightarrow 8$ se izvodi na sledeći način. Umesto u grupe po četiri, podeliti na isti način u grupe od po tri cifre i izračunati oktalni ekvivalent svake grupe čime se dobija cifra po cifra broja u oktalnom zapisu. Za konverziju u obrnutom smeru, obrnuti ceo proces.

2 Opseg tipova podataka u prog. jeziku C

Celobrojni tip	short int	-128 ... 127	1 bajt
	int	-32768 ... 32767	2 bajta
	long int	$-2^{31} \dots 2^{31}$	4 bajta

Prefiks unsigned obezbeđuje samo pozitivne vrednosti pa datom tipu daje opseg $0 \dots 2^n - 1$ za n bajtova.

Float i Double tipovi služe za predstavljanje racionalnih brojeva (realnih brojeva sa ograničenom preciznošću). Zauzimaju 4, odnosno 8 bajtova i čuvaju 6-7, odnosno 14-15 značajnih cifara iza decimalne tačke.

3 Zapis negativnih brojeva, Komplement dvojke

Predstavljanje negativnih brojeva se realizuje preko komplementa dvojke. Formula glasi:

$$N^* = 2^{8 \cdot n} - N$$

gde je:

n - broj bajtova u zapisu

N - pozitivni ceo broj

i N^* u notaciji komplementa dvojke zapisan broj -N.

Praktično ovo znači da će short int sa vrednošću -6 (-N) prikazan kao unsigned short int imati vrednost $N^* = 2^8 - 6 = 250$.

Kada posmatramo binarni zapis, računanje komplementa je nešto lakše. Već pomenuti broj 6 u binarnom zapisu glasi $00000110_{(2)}$. Sada najpre odredimo komplement jedinice (invertovanje bitova) što je u ovom slučaju $11111001_{(2)}$ i zatim dodamo 1. Dakle -6 prikazan kao komplement dvojke glasi $11111010_{(2)} = 250_{(10)}$. Vidimo iz binarnog zapisa da je najveći broj koji na ovaj način možemo zapisati $2^n - 1$, a najmanji -2^n .

4 Konverzija tipova pri evaluaciji izraza

Prilikom izračunavanja nekog izraza, ukoliko su operandi različitih tipova biće izvršena konverzija prema sledećem pravilu:

$$\begin{aligned} & \text{char} < \text{unsigned char} < \text{short} < \text{unsigned short} < \text{int} < \\ & < \text{unsigned int} < \text{long} < \text{unsigned long} < \text{float} < \text{double} \end{aligned}$$

Operand nižeg tipa će biti konvertovan automatski u tip drugog operanda i tek će onda biti izvršeno izračunavanje. Vrednost operanda neće biti trajno promenjena.

primeri:

$-1 < 1U$ je netačno

jer $\text{int} < \text{unsigned int}$ pa će -1 biti prevedeno u unsigned int, a to je po komplementu dvojke 65535.

$-1L < 1U$ je tačno

jer $\text{unsigned int} < \text{long}$, pa će 1U biti prevedeno u 1L i onda stvarno poredimo -1L i 1L.

5 Operatori nad bitovima i operatori pomeranja

« Pomeranje ulevo. $x \ll y = x \cdot 2^y$. Predstavljeno u binarnom obliku ovo zapravo znači pomeranje svih cifara za y mesta ulevo i dopisivanje y nula sa desne strane. Višak cifara se odbacuje sa leve strane. Što se tipova podataka tiče, x mora biti znakovnog ili celobrojnog tipa, a y mora biti celobrojna vrednost.

» Pomeranje udesno. $x \gg y = \frac{x}{2^y}$. Predstavljeno u binarnom obliku ovo zapravo znači pomeranje svih cifara za y mesta udesno i dopisivanje $(y - 1)$ nule sa leve strane. Ukoliko je broj unsigned dopisuje se još jedna nula sa desne strane, u protivnom prepisuje se vodeći bit broja x pre izvršavanja operacije. Opet x mora biti znakovnog ili celobrojnog tipa, a y mora biti celobrojna vrednost.

\sim Negacija nad bitovima. (1 postaje 0 i obrnuto). Promenljiva mora biti celobrojnog ili znakovnog tipa.

$\&$ Konjunkcija nad bitovima. Operandi su celobrojnog ili znakovnog tipa.

$|$ Disjunkcija nad bitovima. Operandi su celobrojnog ili znakovnog tipa.

\wedge Ekskluzivna disjunkcija ($1 \wedge 1 = 0$). Operandi su celobrojnog ili znakovnog tipa.

Postoje i složeni operatori $|=$, $\&=$, $\ll=$, $\gg=$, \dots

6 Prvenstvo operatora

- 1) [LTR] $() [] -> .$
- 2) [RTL] $! \sim ++ -- + - * (\text{type}) \text{sizeof}$
- 3) [LTR] $* / \%$
- 4) [LTR] $+ -$
- 5) [LTR] $\ll \gg$
- 6) [LTR] $< <= > >=$
- 7) [LTR] $== !=$
- 8) [LTR] $\&$
- 9) [LTR] \wedge
- 10) [LTR] $|$
- 11) [LTR] $\&\&$
- 12) [LTR] $||$
- 13) [RTL] $?:$
- 14) [RTL] $= += -= *= /= \% = \&= \wedge= |= \ll= \gg=$
- 15) [LTR] $,$

7 Važnije pozicije u ASCII tabeli

Razmak (Space): 32

'0': 48

'A': 65

'a': 97

Napravljeno uz pomoć L^AT_EX-a